DOCUMENT RESUME

ED 234 768 IR 010 840

TITLE	Frontiers in Educational Computing. Association for Educational Data Systems Annual Convention Proceedings (21st, Portland, Oregon, May 9-13, 1983).
INSTITUTION	Association for Educational Data Systems, Washington, D.C.
PUB DATE	May 83
NOTE	400p.; Best copy available. For related documents, see ED 201 410 and ED 223 239.
AVAILABLE FROM	Association for Educational Data Systems, 1201 16th Street, NW, Washington, DC 20036 ($20.00, per copy).
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - General (140)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; Computer Managed Instruction; *Computer Oriented Programs; Computer Programs; Computers; *Computer Science Education; Elementary Secondary Education; Higher Education; Microcomputers; Programed Instructional Materials; Programing; Programing Languages; Teacher Education; *Technological Advancement
IDENTIFIERS	*Computer Uses in Education; Interactive Systems; LOGO Programing Language

ABSTRACT

The 98 papers in this collection examine a wide variety of topics related to the latest technological developments as they apply to the educational process. Papers are grouped to reflect common, broad areas of interest, representing the instructional, administrative, and computer science divisions of the Association for Educational Data Systems (AEDS), a private, non-profit, international educational organization, which has the aim of providing a forum for the exchange of ideas and information about the relationship of modern technology to education. Included are lists of the 1983 Convention team, AEDS Board of Directors, Affiliate Group Presidents, Institutional Members of AEDS, Sustaining Members of AEDS, and an index of papers by author. Individual papers address such varied topics as the use of microcomputers for drills, tutorials, tests and word processing; developing instructional materials for the computer; modern approaches to learning to program; planning for computer acquisition; computer literacy in a multicultural environment; an attendance system for secondary schools; LOGO in the elementary classroom; programming plans and advance organizers and their uses in improving program debugging performance; setting up a computer program; and micro-assisted school management. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Association For Educational Data Systems

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)
- This document has been reproduced as received from the person or organization originating it.
- Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

AEDS

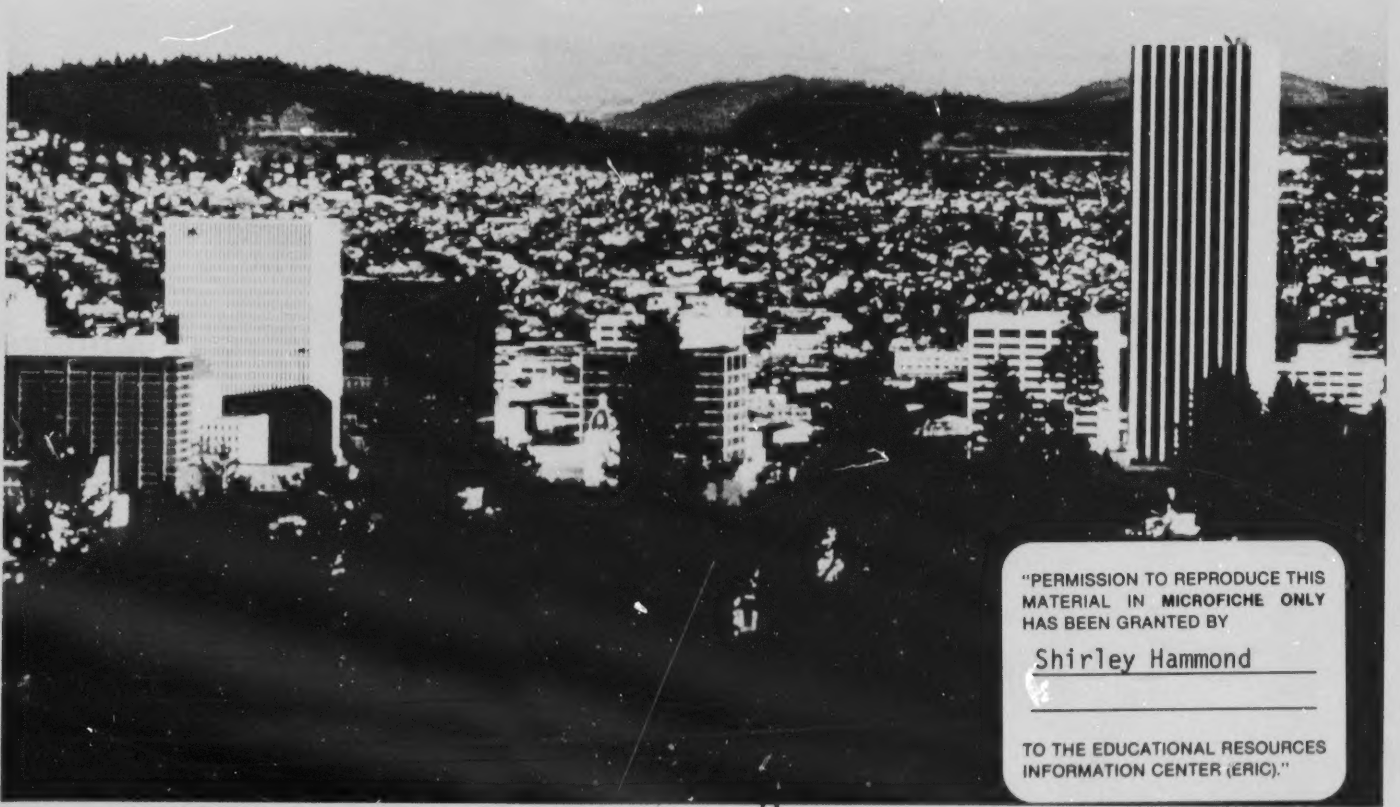
Twenty-First Annual Convention Proceedings

ED234768

"Frontiers in Educational Computing"

May 9 - 13, 1983

Hilton Hotel
Portland, Oregon



"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

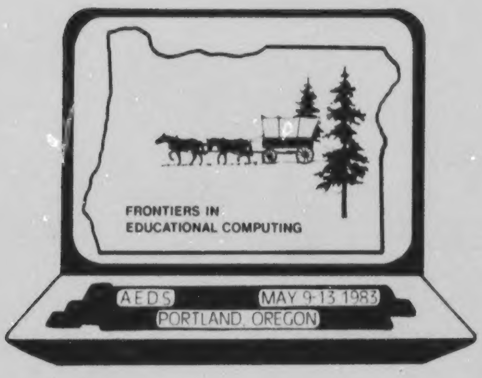
Shirley Hammond

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Additional Copies Available
For $20.00, Prepaid From

Association for Educational Data Systems
1201 Sixteenth Street, N.W.
Washington, D.C. 20036

Library of Congress Catalog Card Number ISSN 0147-9296
Copyright 1983, Association for Educational Data Systems



The cover - -

The city of Portland, Oregon, is situated in the Beautiful Pacific Northwest. Surrounded by mountains, forests, rivers and lakes, it has been acclaimed as one of the most liveable cities in the United States. Famous for its mild climate, revitalized urban areas, cultural centers and friendly people, visitors to the city and surrounding community enjoy an unforgettable experience.

FOREWARD

The Association for Educational Data Systems (AEDS) is a private, non-profit international educational organization founded in 1962 by a group of professional educators and technical specialists in educational application of computers and technology. The aim of the organization is to provide a forum for the exchange of ideas and information about the relationship of modern technology to education.

AEDS membership is open to all persons interested in learning about educational data processing and who want to keep abreast of current developments and directions in educational data systems and computer technology. Included in the membership are educators and technical specialists from all over the United States, Canada, and other countries representing public and private schools, higher education, federal government, the military, and other professional and technical groups interested in the educational use of computers and technology.

These proceedings represent a significant contribution to the growing body of knowledge of the latest developments in technology as it applies to the educational process. Each paper has been selected on the basis of its relevance to the general theme of the convention, "Frontiers in Educational Computing." Papers have been grouped as best as possible to reflect common broad areas of interest representing the instructional, administrative and computer science divisions of the organization. The Convention Committee firmly believes this collection of papers can have considerable impact on the educational community if each paper is given the careful attention it is due.

The Convention Committee would like to express its appreciation for the support and assistance of all of those who have shown an interest in this year's Convention. The AEDS Board of Directors, Oregon Association of Educational Data Systems (OAEDS), the membership, private industry and other interested persons have exhibited interest and initiative which are essential to making the Convention a success.

Most especially, I would like to thank all of the members of the AEDS-'83 Convention Team who have worked so tirelessly on the many details associated with this Convention.

Charles G. Hadduck
Convention Coordinator

TABLE OF CONTENTS

Foreword iii
Table of Contents iv
1983 Convention Team v
AEDS Board of Directors vi
Affiliate Group Presidents vii
Institutional Members of AEDS ix
Sustaining Members of AEDS x
Index of Papers by Author xi
Papers 1

1983 AEDS CONVENTION TEAM

Convention Coordinator

Charles G. Hadduck
Systems Manager
Portland Public Schools
P.O. Box 3107
Portland, Oregon 97208
(503) 249-2000, Ext. 512

Committee Chairpersons

Activities

Porter Sexton
Management Information Specialist
Portland Public Schools
P.O. Box 3107
Portland, OR 97208
(503) 249-2000, Ext. 509

Equipment

Mike Arts
Director, Computer Center
University of Portland
5000 N. Willamette Blvd.
Portland, OR 97203
(503) 287-7317

Evaluation

Paul O'Driscoll
Director Data Processing
Linn-Benton Educ. Service Dist.
905 E. 4th Ave.
Albany, OR 97321
(503) 967-8822

Exhibits

Gerald Larer
Director Data Processing
North Clackamas School District #12
14211 S.E. Johnson Rd.
Milwaukie, OR 97222
(503) 653-3827

Facilities

Dan Moore
Good Samaritan Hospital
1015 N.W. 22nd Ave.
Portland, OR 97210
(503) 229-8100

Finance

Tom Crowder
Director Data Processing
Portland Community College
12000 S.W. 49th Ave.
Portland, OR 97227
(503) 244-6111

Program

Nancy Stiggins
Senior Systems Analyst
Portland Public Schools
P.O. Box 3107
Portland, OR 97208
(503) 249-2000, Ext. 506

Don Holznager
Northwest Regional
Educational Laboratory
300 S.W. 6th
Portland, OR 97201
(503) 248-6800

Publicity

Bob Rintoul
Director Data Processing
Clackamas County ESD
Marylhurst, OR 97036
(503) 635-4341

Registration

Arlen Sheldrake
Director Data Processing
Multnomah County ESD
220 S.E. 102nd
Portland, OR 97216
(503) 777-7425

AEDS BOARD OF DIRECTORS
1982-83

President

Mr. A. V. Majors
Fort Worth Public Schools
Fort Worth, Texas
(817) 336-8691

President-Elect

Mr. John W. Kelly
Palm Beach Junior College
Lake Worth, Florida
(305) 439-8040

Past President

Dr. Dennis Spuck
University of Houston
Houston, Texas
(713) 749-7281

Treasurer

Mr. S. M. Raucher (1984)
Montgomery Co. Public Schools
Rockville, Maryland
(301) 279-3581

Secretary

Mr. Alan Olkes (ex-officio)
Dade Co. Public Schools
Miami, Florida
(305) 552-9894

Executive Secretary

Ms. Shirley Easterwood
AEDS
Washington, D. C.
(202) 822-7845

BOARD MEMBERS

Dr. William C. Bozeman (1984)
University of Iowa
Iowa City, Iowa
(319) 353-5718

Mr. Arthur Daniels, Jr. (1983)
Data Management Solutions
Birmingham, Michigan
(313) 540-2211

Dr. John Hamblen (ex-officio)
National Bureau of Standards
Washington, D. C.
(301) 921-2905

Dr. Kenneth Hasledalen (ex-officio)
MN SD Data Proc. Joint Board
St. Paul, Minnesota
(612) 633-9100

Dr. D. Thomas King (1985)
St. Paul Public Schools
St. Paul, Minnesota
(612) 293-7709

Mr. Rick Meyer (1983)
Maricopa Community Colleges
Phoenix, Arizona
(602) 244-8355

Dr. Lauran Sandals (1983)
University of Manitoba
Winnipeg, Manitoba, Canada
(204) 474-9017

Mr. Lary Smith (1983)
Wayne CO ISD/Data Center
Wayne, Michigan
(313) 326-9300

Ms. Patricia Sturdivant (1985)
Houston ISD
Houston, Texas
(713) 623-5581

Ms. Ann Tompkins (1985)
Westinghouse Information Sys.
Iowa City, Iowa
(319) 354-9200

AFFILIATE PRESIDENTS AND REPRESENTATIVES

FUNCTIONAL

Permanent Contact

Information Processing Administrators of Large School Systems

Alan Olkes (5/81-5/83)
Dade County Public Schools
13135 Coral Way
Miami, FL 33175
(305) 552-9894

Jack Merrill
Memphis City Schools
2597 Avery
Memphis, TN 38112
(901) 454-5425

GEOGRAPHIC

Alaska

Patricia Stowers (4/82-4/83)
AACED
Box 29
Cantwell, AK 99729
(907) 768-2372

Norman Holthouse
Anchorage School District
Pouch 6-614
Anchorage, AK 99502
(907) 269-2251

Alberta

Michael Brebner (7/82-6/83)
Faculty of Education
Education Tower, Rm. 838
University of Calgary
Calgary, Alberta, CANADA T2N 1N4
(403) 284-6495

Arizona

Lenda Helser (5/82-4/84)
Mesa Public Schools
Whittier School
733 N. Lingmore
Mesa, AZ 85201
(602) 833-3899

(HOME): 4316 N. 86th Pl.
Scottsdale, AZ 85257
(602) 945-6691)

California Educational Data Processing Association

Marilyn Carson (11/81-11/82)
Santa Clara Co. Office of Educ.
100 Skyport Drive
San Jose, CA 95110
(408) 947-6517

District of Columbia

Rolin Sidwell (11/82-11/83)
2216 Rand Pl., N.E.
Washington, D.C. 20002
(202) 727-3537

GEOGRAPHIC

Permanent Contact

Florida

Terry Shoultes (11/82-83)
Palm Beach Co. Schools
3323 Belvedere Rd.
W. Palm Beach, FL 33402
(305) 684-5090

Illinois

Ellie Gillespie (11/82-10/83)
DePaul University
64 E. Jackson Blvd.
Chicago, IL 60604
(312) 321-7675

Box 128 (Lyle Smith)
DeKalb, IL 60115

Iowa

Lorraine Patrick (11/82-10/83)
ATECC
500 College Dr.
Mason City, IA 50401
(515) 423-4004

Phil Berrie
AEA II
1932 S.W. 3rd St.
Ankeny, IA 50021

Manitoba

Lauran Sandals (6/81-5/83)
Educ. Psychology
University of Manitoba
Winnipeg, Manitoba, CANADA R3T 3B3
(204) 474-9017

Edward Boorsma
Winnipeg SD #1
1577 Wall Street E
Winnipeg, MB, CAN R3E 2S5
(204) 775-0231

Maryland

Frank M. White (4/82-3/83)
Dept. of Comp. DP Tech.
Catonsville Comm. College
Baltimore, MD 21228
(301) 455-4443

Michigan

David Rhodes (10/82-10/83)
Berrien Co. Indep. School Dist.
711 St. Joseph Ave.
Berrien Springs, MI 49103
(616) 471-7725

Mid-America

James Wenger (6/82-6/83)
Emporia State Univ.
1200 Commercial
Emporia, KS 66801
(316) 343-1200 X488

GEOGRAPHIC	Permanent Contact
Mid-South	
Barbara Straka (3/82-2/84) Dept. of Computer Science Jackson State University Jackson, MS 39217 (601) 968-2105	
Minnesota	
Sue Gerk Talley (7/82-6/83) TIES 1925 W. Co RD B2 St. Paul, MN 55113 (612) 633-9110	
Nebraska	
Susan Manuel (6/82-5/83) Educ. Svc. Unit No. 3 4224 South 133 Street Omaha, NE 68137 (402) 330-2880	
New Jersey	
Harry Selover Asst. Supt. Newton Public Schools 57 Trinity St. Newton, NJ 07860 (201) 383-1900 X502	Irvin Gaydos State Dept. of Education 225 West State Trenton, NJ 08601 (609) 292-6037
New York	
Donald Ross (11/81-11/83) Ardsley High School 500 Farm Road Ardsley, NY 10502 (914) 693-6300	
North Carolina	
Margaret Bingham (3/82-3/83) NC Dept. Public Instruction 116 W. Edenton St. Raleigh, NC 27611 (919) 733-4695	Sammy Small (Treasurer) Broughton High School 723 St. Mary's St. Raleigh, NC 27605 (919) 755-6226
Ohio	
Tom Stellers (7/82-7/83) Mahoning Co. Board of Education 25 E. Boardman St. Youngstown, OH 44503 (216) 746-2207	

GEOGRAPHIC	Permanent Contact
Oregon	
Mike Arts (1983) University of Portland 5000 N. Willamette Blvd. Portland, OR 97203 (503) 283-7317	Charles Hadduck Portland Public Schools
Texas	
Bob Goss (10/82-10/83) Computing Services Southwest Texas State University San Marcos, TX 78666 (512) 245-2501	Iva Helen Lee McLennan Comm. College 1400 College Drive Waco, TX 76708 (817) 756-6551 X353
Virginia	
G. Larry Brown (7/82-6/83) Rte. 3, Box 368 Charlottesville, VA 22901 (804) 977-3900	James Augustine Norfolk City Pub. Sch. 800 E. City Hall Ave. Norfolk, VA 23510 (804) 441-2458
Wisconsin	
Rachelle Kolbus (7/82-6/83) Racine Unified School Dist. 2220 North Western Ave. Racine, WI 53404 (414) 631-7116	Rolland Hicks Dir. NW Comp. Support Ctr. CESA #6 West Park Ave. Chippewa Falls, WI 54729

Geographic Affiliate Representative

Kenneth Hasledalen (5/82-6/83)
MN SD Data Proc. Joint Board
1925 W. County RD B2
St. Paul, MN 55113
(612) 633-9110

Geographic Affiliate Rep-Elect

Frank White (6/83-5/84)
Dept. of Comp. DP Tech.
Catonsville Community College
Baltimore, MD 21228
(301) 455-4443

INSTITUTIONAL MEMBERS

Adams County School District #12, Northglenn, Colorado
American Samoa Government, Dept. of Education, American Samoa
Aramco Schools, Dhahran, Saudi Arabia
Auburn University, Auburn, Alabama
Baltimore City Public Schools, Baltimore, Maryland
Berkeley County Board of Education, Martinsburg, West Virginia
Boulder Valley Public Schools, Boulder, Colorado
Broward County School Board, Ft. Lauderdale, Florida
Buffalo Public Schools, Buffalo, New York
Brigham Young University, Hawaii Campus, Laie, Hawaii
Cecil County Board of Education, Elkton, Maryland
Cincinnati Public Schools, Cincinnati, Ohio
Coast Community College District, Costa Mesa, California
College of DuPage, Glen Ellyn, Illinois
Cuyahoga Community College, Cleveland, Ohio
Downers Grove High School North, Downers Grove, Illinois
East Texas State University, Commerce, Texas
Eastern Kentucky University, Richmond, Kentucky
Fairfax County Public Schools, Fairfax, Virginia
Fordham University, Bronx, New York
Georgetown University, Washington, D. C.
Hastings Public Schools, Hastings, Nebraska
Hillsborough County Public Schools, Tampa, Florida
Howard County Public School System, Ellicott City, Maryland
Hutchinson Community Junior College, Hutchinson, Kansas
International Institute of Applied Technology, Inc., Germantown, Maryland
Jackson Municipal Separate School District, Jackson, Mississippi
Jackson State University, Jackson, Mississippi
Janesville School District, Janesville, Wisconsin
Jefferson County Public Schools, Lakewood, Colorado
Kansas City Kansas Community College, Kansas City, Kansas
Kansas State Department of Education, Topeka, Kansas
Lake Washington School, Kirkland, Washington
Mahoning County Office of Education, Youngstown, Ohio
Mankato State University, Mankato, Minnesota
Maricopa Community Colleges, Phoenix, Arizona
Maryland State Department of Education, Baltimore, Maryland
Metro II, St. Paul, Minnesota
Miami-Dade Community College, Miami, Florida
Minnesota Educational Computing Consortium, St. Paul, Minnesota
Minnesota School District Data Processing Joint Board, St. Paul, Minnesota
National Institute of Technology, Cross Lanes, West Virginia
New York State Education Dept., Albany, New York
Northwest Regional Education Laboratory, Portland, Oregon
Oklahoma City Public Schools, Oklahoma City, Oklahoma
Oral Roberts University, Tulsa, Oklahoma
Orleans Parish School Board, New Orleans, Louisiana
Palm Beach County School Board, West Palm Beach, Florida
Palm Beach Junior College, Lake Worth, Florida
Rosemount Independent School District #196, Rosemount, Minnesota
Rutgers University, CCIS-CRC, New Brunswick, New Jersey
Saipan Department of Education, Saipan, CM
San Mateo County Schools, Redwood City, California
Sault Ste. Marie Area Public Schools, Sault Ste. Marie, Michigan
Shawnee Mission Public Schools, Shawnee Mission, Kansas

Strayer College, Washington, D. C.
Thomas Jefferson University, Philadelphia, Pennsylvania
Topeka Unified School District 501, Topeka, Kansas
University of Alberta, Edmonton, Alberta, Canada
University of Delaware, Newark, Delaware
University of Iowa, Iowa City, Iowa
University of Pennsylvania, Philadelphia, Pennsylvania
U.S. Naval Academy, Annapolis, Maryland
Valencia Community College, Orlando, Florida
Wayne County Intermediate School District, Wayne, Michigan
Wichita Public Schools, Wichita, Kansas

SUSTAINING MEMBERS OF AEDS

DIGITAL EQUIPMENT CORPORATION, Marlboro, Massachusetts
NATIONAL COMPUTER SYSTEMS, Minneapolis, Minnesota
TIME SHARE CORPORATION, Hanover, New Hampshire
WESTINGHOUSE INFORMATION SERVICES, Iowa City, Iowa

PAPERS BY AUTHORS

ANAGNOST, JOHN T.
The Impact of the Microprocessor on the Work Force and the Necessity for New Programs in Engineering Technology 1

ARGANSKI, NICHOLAS
Use of Micros for Drills, Tutorials, Tests and Word Processing 6

BAUGHMAN, JO ANN
Developing Instructional Materials for the Computer 9

BAUGHMAN, JO ANN
Why Conduit? 14

BERRIE, PHILLIP J., Ph.D.
"Why Teach About Computers?" 18

BLANKS, EDWIN E.
Structured Systems Techniques Used in Long Range Planning for Administrative Educational Systems 21

BORK, ALFRED
Modern Approaches to Learning to Program 24

BORK, ALFRED
Research and Development at the Educational Technology Center 26

BREBNER, ANN
Equipping Teachers for the Unfinished Revolution: Some University Courses on Computer Education 29

BRIGHT, GEORGE W.
Diagnosis Subroutines in Drill and Practice Programs 33

BRITTAIN, DAVID L.
The Florida Information Resource Network (FIRN) 37

BROGAN, RAYMOND F.
Planning for Computer Acquisition 41

BROWN, DANIEL J.
Reactions to the Administrative Use of Microcomputers 45

BRUCE, ROBERT
A Telidon - Apple II Instruction Module 50

BULLER, EDWARD, B.Sc., B.Ed., M.Ed.
BURRON, DONA, B.A., B.S., M.Ed.
LOGO K-6: The development of Thought and Meaning 53

BURNS, GEORGE
Children's Approaches to Programming: A Teacher's Perspective 55

CAIN, RALPH W.
SEMMES, PATRICIA A.
The Use of Logo in the Teaching of Geometry K through 12 59

CAREY, ERNEST
Computer Literacy in a Multicultural Environment 63

CARLAND, JoANN C.
CARLAND, JAMES W.
Computer Applications to Testing/A Microcomputer Examination Support System 66

11

CARTER, JAMES A., JR.
Computers in Industrial Education: An Often Forgotten But Very Fertile Field 69

CLANCY, ALLEN
Foreign Language Computer Lab 73

CONNOLLY, FRANK W.
Selecting Micros is a Sequential, not a Random, Activity 76

COOK, JOHN B., Ph.D.
Adventures in Education: Some Uses for Computerized Adventure Games in Education 80

DORSEY, WILBUR C.
An Attendance System for Secondary Schools 83

EDWARDS, LOIS
Problem Solving Courseware in Advanced Mathematics Classes 86

EMERY, NEAL T.
The Ontario Perspective 90

FERRET, JOHN J., Jr.
"LOGO in the Elementary Classroom: A Learning Environment" 96

GODDARD, WILLIAM P.
Bach to Basic (CAI, CAL, CMI, Admin., Teacher Ed., BASIC, LOGO, PILOT, PASCAL) 100

GRESS, EILEEN K.
Administrative Planning and Implementation for Instructional Computing 104

GRIFFITH, DAVID W.
Word Processing for the Junior High Student with Weak Writing Skills 108

GRISWOLD, PHILIP A.
GRISWOLD, MARGARET MILLER
MOBLEY, GORDON
Differences in Computer Awareness Between Education and Business Majors 112

HAMOLSKY, MONICA C.
"Effective Teacher-Trainer Programs: Of What Do They Consist? An Examination of the Research" 116

HAMPSHIRE, GLEN
Computers are Here 120

HANSEN, DR. DENNIS C.
BAXTER, CONNIE
Caution: Your Garden of Eden has an Apple II 125

HARTMAN, JANET D.
Creating Instructional Materials for the Microcomputer 128

HASSELL, JOHNETTE
Programming Plans and Advance Organizers and Their Uses in Improving Program Debugging Performance 131

HEMPHILL, BARRY R.
The Information Center - End-user Computing in High Education - A New Direction 137

HOYLE, R. KEITH
Microcomputers: Psychological Reporting, Data Base Management, 94-142 Implementation .. 141

JACKSON, ROBERT W.
Setting Up a Computer Program: Now That You Have a Computer, What Are You Going To Do With It? 143

JELDEN, D. L.
Simplified Tutorial Programming for Interactive CAI 147

JUDD, DOROTHY H., Ed.D.
Microcomputers at Home: Potential for Classroom Instructional Support 152

JUDD, ROBERT C., Ph.D.
The Microcomputer: Analytic Tool for MBA Students 156

KAUFFMAN, FRED H.
Computer Networking at Broward County - A Status Report 160

KEHRER, GARY A.
SCHEPIS, NICK A.
The Twentieth Century Frontier: From Blackboard & Chalk to Monitor & Keyboard 165

KERR, THOMAS
Computer Awareness Magnet at Evans 169

KING, JAMES E.
Microcomputer Selection 173

KISER, CHESTER, Ed.D.
Strategic Planning of School District Small Computer Systems 177

KUCHINSKAS, DR. GLORIA
Interactive Language Experiences Between the Computer and the Student 181

LEE, SEONG-SOO, Ph.D.
A Learner-Based Computer Instructional System: An Experiment 185

LORTON, PAUL
Law and Order on the Micro Frontier: A shared disk system for educational microcomputers 189

LUND, CHARLES
The Design and Use of Computer Learning Games 194

MAFFEI, ANTHONY C.
A Successful Technique for Introducing **BASIC** in the Schools 199

MARSHALL, JON C.
Using Machine Readable Surveys in Education: A Practitioner's Perspective 201

MC KYE, GORDON G.
Computers in Education - Software Cataloguing Project 205

MILLER, SAMUEL K.
Getting Started with LOGO: A Direct Instruction Approach 209

MODESITT, KENNETH L.
Computer Based Learning: Important Problems, Creative People, and Powerful and Affordable Tools 213

MURPHY, SUZANNE
Project Disco 218

NEELAN, IAN
Geoprocessing as Applied to an Integrated Data Base 222

NICOLSON, NORMAN
Economic Issues in the Management of Instructional Computing Resources 226

PAGE, BRUCE
Staff Training for Successful CAI 230

PHILLIP, CHARLES C.
Identifying Hardware for High School Microcomputer Laboratories 235

PICKLES, BOB
"The Computer as a Classroom Tool" - A Statewide Inservice Project 239

PITTS, GERALD N.
EGGEN, MAURICE L.
CAI Courseware Development Through Interactive Prompting 242

PROSEN, TOM
A Programming Language for Elementary Students 247

RAMQUIST, DR. RAYMOND
Preparing Future Teachers for a Computer School 251

RAUCHER, STEPHEN M.
KOEHLER, T. J.
Competency Based Training of Software Specialists 255

RIVERS, ROBERT
VOCKELL, EDWARD
Using Computer Simulations to Enhance Problem Solving Skills in Science 262

RODEL, LEE J.
A Generic Instructional Management System Developed by Educators for Educators 266

ROSEN, BERNARD; FRIEDMAN, EDWARD A.; JURKAT, M. PETER; MOELLER, JOSEPH J. JR.;
PINKHAM, ROGER S.
Personal Computer Plan at Stevens Institute of Technology 270

RUSHINEK, SARA
RUSHINEK, AVI
Plan for Middle Management and Its Impact on Decision Support Systems 272

SEMMES, PAT
WILSON, JEAN
Computers in Education: Booting up a University 276

SEXTON, PORTER
A Decision Support System for Determining Equality of Educational Opportunity in a Public School System 279

SHAVELSON, RICHARD; WINKLER, JOHN; STASY, CATHLEEN; ROBYN, ABBY
Teachers' Instructional Uses of Microcomputers 283

SLOAN, SALLY A.
This Way to the Bar 287

SMALT, STEVEN W.
TRIGG, RODGER, Ph.D.
The Computer - A Great Role in Accounting Education? 291

SMITH, LYLE B.
A Computer Science Curriculum That Isn't The Same 295

SMITH, LYLE B.
GC101: A First Course in Computer Graphics 300

SMITH, VICKI
Software Evaluation: Support Through the Texas Education Computer Cooperative 304

SOUDER, RAY H.; LEIGH, WILLIAM; HOLLOWAY, WILLIAM
Using Exploratory Data Analysis to Evaluate Student Performance in the Classroom 307

SPEIKER, CHARLES A.
MANUEL, SUSAN L.
Micro-Assisted School Management - A Training Strategy that Works 312

STANGER, DONNA
Problem Solving Skills and Strategies: A Generic Matrix for Elementary Computer Assisted Instruction 316

STAPLETON, C. G., Ph.D.
Special Teacher Use of Microcomputers for the Writing of Individual Educational Plans 321

STIER, MARGARET F.
The Rationale for Inclusion of Computer Literacy Within Developmental Reading Classes 324

SZABO, MICHAEL, Ph.D.
Improving Computer Based Learning Course Through Application of Cognitive Science Learning Theory 328

THOMAS, GREGORY P.
WOLMUT, PETER
Computer Based Aids for Curriculum Planning 333

TOMBOULIAN, NOLAN
Effective Use of a Micro-Based File Management System 337

TOOP, JUDITH L.
Computer as Tutor 341

TURNER, GREG
Microcomputer Applications in the Special Education Classroom 344

TURRIETTA, ANTHONY
TURRIETTA, CHERYL
Computer Music Application in Education: The Soundchaser Computer Music System 346

VINSON, JAMES S.
Computer Literacy Programs and Faculty Development 350

VOLDING, GARY D.
ICLS - Integrated Classroom Learning System 353

WHOLEBEN, BRENT E., Ph.D.
Tort Liability for Schools in the use of Microcomputers as Instructional Devices in the Classroom 355

WHOLEBEN, BRENT E., Ph.D.
Self-Sufficiency Approach to Computer Literacy Training for Private and Parochial Schools 360

WIDMER, CONNIE CARROLL
PARKER, JANET
In-Service Computer Workshops to Maximize Enthusiasm and Minimize Apprehension 364

WILDE, JANE H.
Education Computing in Southeastern States 368

WILSON, RAY
Microcomputers in Geographically Isolated Areas 372

WRIGHT, PETER W.
Managing Micromania and Beyond 376

ZUCKERMAN, ROBERT A.
Computers in Special Education: A Model for Making the Match 380

THE IMPACT OF THE MICROPROCESSOR ON THE WORK FORCE AND THE NECESSITY FOR NEW PROGRAMS IN ENGINEERING TECHNOLOGY

by

John T. Anagnost
Department of Research
Bell & Howell Education Group
2201 W. Howard, Evanston, IL 60202

It is the purpose of this paper to indicate the impact of microelectronics applications on employment in the 1980's and how the new technologies impact Technician and Technologist Programs in educational institutions. Concerns have been expressed by governments and labor organizations and studies conducted how the introduction of the microprocessor, with its applications in Numerical Control (N/C) and Robotics, will cause wide-spread disruption in the labor market.

Keywords: Microelectronics, Chips, Engineering
Technology, Microprocessor

1. Introduction

A new technology, which can replace human beings in industry, may be expected to significantly impact society as a whole.

There is now mounting evidence that this introduction of new technologies may cause people displacement rates estimated in robots alone, by the Carnegie Mellon University Studies (November 1980), of 4,000,000 people by 1990; and in a separate study conducted by the Eikonix Corporation (September 1979), as much as 6,000,000 in this one application alone (United States), /1,2/. Education institutions must concern themselves with adopting new programs in technology to help industry operate, maintain and help in the design of automatic devices.

1.1 Chip Stories

Generally, microelectronics technology began in 1946, with the invention of the bipolar junction transistor, which led to the development of integrated circuits and advances in miniaturization. The techniques involve putting a number of circuit components on thin copper boards and using photographic and other methods to scale the boards down to the size of "chips". Chips now have component circuits separated by distances on the order of one micron (one millionth of a meter).

As an expample, the Japanese have developed a chip mounted on a 40-pin, ¼-inch square that contains 200,000 components with spacings of less than one micron between lines. It has been estimated that the lower limit of spacing is less than 5/100 of a micron.

Microprocessors are, in effect, control devices which increase the efficiency of a technical process or the flexibility of a product in which they are used. Computer-aided manufacturing and word processors are examples.

Familiar applications of the microprocessor can be found in such consumer goods as radios, television sets, pocket calculators, watches, and others. Microprocessors are also carrying out a growing range of industrial operations, such as the tasks involved in counting and weighing, production control, stock counting and control, and management information processing.

1.2 Robots for All Seasons

Of particular interest is the microprocessor's capability in regulating a sequence of operations in numerically controlled machines or robots. The printing, publishing, and telecommunications industries have already been profoundly affected by this technology. In the next decade, microprocessors are expected to have dramatic impact on banking and finance in the transfer of funds, in checking, while personnel departments will be able to use chip technology in processing job descriptions, vacancies, and employee files.

Particularly susceptible to automation are clerical and administrative jobs that involve collation and transcription of large amounts of information, e.g., data preparation, inventory control, and production control. Affected jobs include clerks, program chasers, typists, and secretaries /3/.

In Manufacturing and the metals industry, the impact of robotics and numerically controled machines is expected to be similarly great. Table I shows estimates of the numbers of workers in certain areas of manufacturing who can be replaced by robots. Estimates of displacement figures range from 4 million to 6 million by 1990./1,2/

Table I

Estimates of the Number of Jobs Robots Could Do in Dey Manufacturing Areas

	Total Employed 1979	% of Jobs Which Could Be Done By		No.of Jobs Which Could Be Done By	
		Level I	Level II	Level I	Level II
Assemblers	1,289,000	10	30	128,900	386,700
Officers, Examiners and Inspectors	746,000	15	35	111,900	261,100
Packers and Wrappers (except meat and produce)	626,000	15	40	93,900	250,400
Production Painters	185,000	45	65	83,250	120,250
Welders and Flamecutters	713,000	25	50	178,250	356,000
Machinists and Machine Operators	3,027,000	20	50	605,400	1,513,500
TOTALS	6,586			1,201,600	2,887,950

(Note: Tactile sensing and vision distinguish Level II robots from those of Level I. Level II robots are still in the formative stage.)

1.3 Robots for All Seasons

In the final analysis, the efficiency of any company depends on the quality of education and training of its people. Their productivity is likewise determined by working conditions, degree of job security, commitment to company goals, and the ability of management to motivate them. The Japanese, by virture of industry commitment to job security, have developed a work force that accepts change in technology. While companies are willing to accept lower profits rather than treaten job security, the number or robots in Japan is nevertheless larger than in the rest of the world combined (see Table II). /5,6/

Table II

Japan	14,000
United States	6,200
West Germany	1,200
Sweden	600
Poland	360
Great Britian	185
Norway	170
Finland	110
Denmark	85
Netherlands	35

Table III

Western Europe: Indications Of The Impact Of Microelectronic Applications On Employment in the 1980's /3/

Sector	Employment 1977 (Millions)	Past Employment Trend	Degree of Impact Microelectronics On Sector	Of Employment Trend In the 1980
Agriculture	23.5	↓	Low	↓
Minting	1.2	↓	Low	↓
Manufacturing	40.4	→	High	↓
Utilities	1.4	→	Low	→
Construction	11.5	→	Low	→
Distribution, Hotels Catering	22.2	→	Low	↑
Transprot & Communications	9.7	→	High	↓
Finance	7.9	↑	High	↓
Community & Personal Services	28.9	↑	Low	↑
Other Services	.5		Low	

Some projected unemployment figures by selected countries are shown below:

Table IV

Loss In Jobs By 1990 Or Unemployment Rate

	Percentage
Canada	20% Unemployment
United States	14% Unemployment
United Kingdom	16% Displacement of Labor
France	18% Unemployment
Federal Republic of Germany	40% Estimated Unemployment
Japan	8% Unemployment

Source: Various O.E.C.D. Reports and the U.N. Statistical Year Book, 1978.

2.2 The Necessity For New Programs In Engineering Technology

The preceeding Table provides strong evidence that wide-spread displacement and unemployment will result from the applications of the microprocessor. Educational institutions and companies must concern themselves with adopting new programs which will enable industry to operate, maintain, and in some instances, help in the design or some of these automatic devices. Some schools have already anticipated the wide-spread uses of the computer in their Technology Programs. However, it should be noted that the dissemination of such programs so far has been extremely limited. In addition, implementation of such programs also helps to alleviate problems caused by unemployment, either retraining those displaced or alternately providing a work force which can be quickly employed. There is now considerable evidence that some categories of information workers, computer scientists, managers, computer analysts, programmers and hardware computer technology personnel, will need to be trained in these new areas.

2.3 The Technologist Curricula

The Accreditation Board for Engineering Technology (United States) defines engineering technology as being primarily concerned with the application of established scientific, engineering knowledge and methods /7/. Normally, engineering technology is not concerned with the development of new principles of methods. An engineering technologist is on whose education and experience qualify him to work in the field of engineering technology. He is different from a technician or craftsman in that he has knowledge of scientific, engineering theory and methods, and uses his technical skills in support of engineering activity. Any technology curriculum can be thought of as a planned sequence of college level courses which can lead to either an associate or bachelors degree which is designed to prepare students to work in the field of technology. Readers are cautioned to remember the distinction between quality and level: A very low level course such as dish washing, can be very high quality in the sense that it completes its objective, e.g., and excellent dish washer. On the other hand, courses in mathematical numerata may be thought of as being extremely high level but may be of low quality if the student has not learned the objectives of the course.

2.4 General Considerations

Generally, Associate Degree Programs are two academic years or 60 credit hours and ABET recommends a minimum of 120 semester hours for the Bachelors Degree. Our own experience shows that longer than this is needed to educate a technician or technologist and we now offer 120 credit hours towards an Associate Degree and 154 hours for a Bachelors Degree. Technical courses should be quantitative and textbooks used by developers should contain the necessary mathematical levels compatible with the objectives of the program. Physics should be taught separate from the major. There should be a minimum of at least one (1) year with associated labs. It is imperative that, in Technology Programs, mathematics be taught to include both differential and integral calculus, applied differential equations probability and statistics and Boolean algebra. Such courses can be taught separately or topically at college levels by faculty qualified in these areas and familiar with engineering technology objectives. Curricula should be arranged and taught prior to the time that they are needed in the technical laboratories. Communications and technical writing skills for technologists are absolute musts. Any curriculum design should include a minimum of at least nine hours of communications. General education studies are important to the cultural background of the technologist. This background will help determine some of his attitudes towards life, his ability to get along with his peers, his subsequent activity in the community in which he lives and with it, being functional in a business environment. Finally, laboratories should be equipped and staffed and designed in such a way as to demonstrate the theory learned in the classroom. Nothing is more devastating to the student than to go into a laboratory and encounter a lab experiment that is not related to what had been taught in the classroom. Engineering technologists work in laboratories in support of engineering projects as well as in field service and other areas with which developing countries will be concerned in the future.

2.5

Recently, the Institute of Electrical and Electronics Engineers (I.E.E.E.) Computer Society, conducted a study resulting in the publication "Draft Report on Computer Systems Technology on Education" /8/. The reason for the study was to anticipate the impact on technological education due to microprocessors and the rapid expansion of automatic devices. Figure V shows the generalized subject matter recommended. Clearly, there are more topics here than can be included in a standard 2-year program and this provides an opportunity for those to select courses they feel most indigenous to their own areas of interest. Emphasized are hardware, in the form of computer peripherals, microcomputers, fundamentals of electronics and digital logic, programming, operating systems and mathematics which are combined with laboratory skills and the communications skills necessary for the technician.

Table V

Associate Degree Comparisons
Curriculum Summary In Semester Credits

Basic Science Course	ABET Recommendations	BHEG		U.S. Community Colleges	
Mathematics		23		10	
Algebra					
Trigonometry					
Calculus					
Differential Equations					
Transform Analysis					
Physical Sciences	15	9	32	4	14
Non-Technical Courses					
Communications		10			
English Composition					
Speech					
Report Writing					
Humanistic-Social Studies				24	
Economics					
Literature					
Psychology					
History of Technology					
Other	8	6	16	3	27
Management					
Human Relations					
Or Additional Humanistic- Or Social Studies					
Technical Specialties	30	72		32	
Program Integration And Local Needs (ABET)	7				
TOTALS	60	120		73	

Our own Electronics Technician Program emphasizes these particular areas. This program is qualitative as against being quantitative and does not go into computer system technology in the same depth as the I.E.E.E. Study. This is particularly true in the areas of mathematics, communications and general education. However, there is a strong emphasis on hardware and applications of digital electronics to instrumentation and control, and courses in microprocessors. Such programs are designed with the technician in mind. Generally, a technician operates, maintains and installs electronic equipment.

2.6 Programs In Engineering Technology

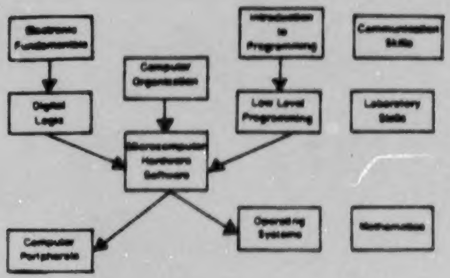
Due to the impact of computers in electronics design and its impact on systems, the Bell and Howell Education Group has restructured its Electronics Technology Program. The result: A nine-term program leading to a Bachelors Degree. (If a student wishes, he can obtain an Associate Degree in seven terms.) The curriculum places strong emphasis on integrated circuit applications, microprocessor hardware and software, digital/data communications, digital control systems and computer programming.

3.0 Conclusions

The accelerated use of the microprocessor in such industries as manufacturing, banking and consumer goods will result in increases in productivity. The use of chips in numerically controlled machines, robots and other applications will also, however, cause people displacement with resultant unemployment. Educational institutions will play a key role in softening social impact involved by providing and retraining of displaced persons and an available work force as the new technologies expand. Needed will be programmers, systems analysts, computer scientists, electronics technicians and technologists. Developing countries may wish to begin with qualitative programs for technicians to operate, maintain and install equipment and then proceed to Technology Programs as sophistication in microprocessor technology develops.

Figure I

SUBJECT MATTER AREAS IN COMPUTER SYSTEM TECHNOLOGY



CURRICULUM CONTENT

Software - Oriented Content:

A. Introduction to Programming
   Module 1: History of Computers and Programming
   Module 2: High Level Language Concepts
   Module 3: The Computing System
   Module 4: Data Representations

B. Computer Organization
   Module 1: Review of Data Representation
   Module 2: Computer Units
   Module 3: Instruction Formats
   Module 4: Microprogramming Level

At the entry level, people trained as service technicians may find jobs:

- In the field, at customer sites.
- In vendor repair depots.
- In in-house installations at the computer manufacturer's own facilities.
- With 'third party' maintenance firms; but the norm still is that such firms require experienced personnel to stay competitive with one another and vendor service groups.

BIBLIOGRAPHY

1. Carnegie-Mellon University, "The Impact of Robotics on the Work Force and Work Place", June 1981

2. Technology Assessment "The Impact of Robotics," Eikonix Corporation, September, 1978

3. "The Impact of Microelectronic Technology on Employment" prepared for Labor Market Research Unit, Ontario Manpower Commission, July 1980

4. "Projections for the Need for Electronics Technicians and Technicians and Technologists in the U.S., "Electronics Industries Association, 1981

5. U.S. News and World Report, "When Robots Take Over People's Jobs," February 1981

6. Business Week, "Robots Join the Work Force," June 1980

7. Standards for Engineering Technology, "Accrediting Board for Engineering and Engineering Technology," 49th Annual Report, September 1981

8. "Two Year Computer System Technology Curricula for the 80's," Donald N. Palko and David M. Hata, T.H.E. Journal, March 1982

USE OF MICROS FOR DRILLS, TUTORIALS, TESTS AND WORD PROCESSING

Nicholas Arganski
John Abbott College
Department of Mathematics and Computer Science
Ste. Anne de Bellevue, Quebec H9X 3L9

ABSTRACT

Even though this paper is aimed at teachers of Calculus, some of the systems presented can be used in other disciplines. For example, the Test Generating System and the System for Individual Supplementary Instruction do not need any additional features to enable them to handle any other discipline in natural or social sciences. The systems discussed are mainly based on CP/M and Microsoft BASIC; they are well documented and are presently partially executable on the NorthStar Advantage (64K).

INTRODUCTION

The H.P. 3000 version of the main system discussed, the Individual Derivative Solving System, were originally presented at the twentieth annual ADES convention (see pp. 122-125 of the proceedings), under the title Calculus Tutorial System - (CTS).

The aim of the CTS is to:
a) assist the student by providing him with drills in basic concepts (Derivative and Integral Solving Systems);
b) help him to assess his comprehension of the subject via tutorials (Individual Supplementary Instruction System);
d) assist the instructor in the evaluation process of the student (Test Generating and Evaluating System);
e) give the instructor some hints for use of the WordStar word processing system;
f) provide the student with an alternative text book on introductory Calculus.

DRILLS

The Derivative and Integral Solving Systems are based on the idea that differentiation is recursive in nature, i.e.

$$d/dx\ u(v(x)) = u'(v(x)\ dv/dx$$

Therefore, the focal point of the system is the recursive algorithm of the Interpreter module.

The function entered by the user is split into separate terms and each term is sent to the Interpreter. The Interpreter identifies the term and sends it to the appropriate module. The module differentiates the term, provides explanations, if requested, and returns the result to the Interpreter which then sends it to the next module in the recursive chain, or, if the differentiation is complete, to the Simplification module.

For example, Sin(e^{x^2}) would first be processed by the trigonometry module, then by the exponentials module and, finally, by the power rule module.

The system has the following features:

a) While entering the function, the student is assisted by the "blinking balancing brackets" function.

b) The original function, the summative result and the current rule involved are displayed at the top of the screen.

c) In the "talk mode" the student has a choice of three levels of explanation.

He has also the following functions at his disposal:

1. Change current drive.
2. Go to beginning of line.
3. Break.
4. Delete character.
5. End program.
6. Go to finish of line.
7. Retrieve previous function.
8. Insert character on/off.
9. Search forward for "X".
10. Search backward for "X".
11. Character left.
12. Output explanation on print.
13. Suspend output.
14. Output solution on printer.
15. Character right.
16. Suspend execution.
17. Change talk mode to level X.
18. Change independent variable.
19. Change order of derivative.
20. Restart IDSS.
21. End program, return to CP/M.

TUTORIAL

Addition of graphic illustrations constitutes the main improvement in the Individual Supplementary Instruction System.

GRAPHICS

The Calill system has the following features:

1) Draws graphs of algebraic or transcendental functions.

2) Illustrates the idea of tangent to a curve by drawing a sequence of chords.

3) Illustrates the idea of the area under a curve by drawing a sequence of rectangles.

4) The idea of volume of a solid is illustrated, either by a sequence of disks, or shells.

The graphics can be generated on the screen of a NorthStar on transparency or paper using the HP 7470A plotter, or the screen contents can be damped on paper using the Integrated Data Systems Prism printer.

TESTS

The editing and the printing facilities were the weakest points of the Computer Generated and Evaluated Problem Sets, the test generating system of the H.P. 3000 version of the CTS. The microcomputer version uses WordStar for editing of problem files and Decwriter III, or Integrated Data Systems Prism for printer.

The original system was designed to utilize the subscript and superscript functions available on the Diablo printer. This was a very slow process. Therefore, we were happy to discover that the subscript and superscript functions available on the WordStar could be used in conjunction with the Decwriter III, which was not meant to provide such service.

To get this combination to work, one has only to select the proper line-spacings on the two instruments. They are as follows:

WordStar

Set line space to 3
Set page length (.PL) to 132

Decwriter III

Set vertical pitch to 12

Try this and you will see for yourself how well it works.

You can also experiment with time-consuming graphics like zigmas, integrals, and other complicated print expressions. Once constructed, these symbols can be stored in a "utility" file and then transferred to your current file using the "Move Block" function of the WordStar.

Stop Press

At press time, only the Individual Integral Solving System and the Calill System are operational.

ACKNOWLEDGEMENTS

In the development of this project I have been assisted by many outstanding students at John Abbott College. Here are their names in chronological order: Larry Barth, Colin Cameron, Glenn Connery, Scott Darlington, Loulou Fitzgerald, Geoffrey Funnel, Andy Gauthier, Geoff Hendren, Ralph Kroll, Dave Mackie, Rick Martin, Robert Parnigani, Ken Robertson, Ronald Shutler and Irving Witte.

DEVELOPING INSTRUCTIONAL MATERIALS FOR THE COMPUTER

Jo Ann Baughman
Oregon State University
Corvallis, Oregon 97331

ABSTRACT

After review of a fair number of CAI authoring guides and authoring languages[1], I have concluded that a dimension of authoring is lacking. That dimension is the application of an instructional theory to the material that is coded. What present authoring guides do give us is a set of rules for making the computer more "user" oriented.

While these rules are important, and necessary to avoid frustrating computer learners, it is very possible to follow them to the letter and produce ineffective instruction.

To develop quality instructional computer materials, you need to consider the instructional design, the instructional strategy, and the interactive style.

INSTRUCTIONAL DESIGN

Models for systematic approaches to designing instructional materials abound. Andrews and Goodson[2] identified over 60 such models and went on to produce a comparative analysis of 40 representative models.

Some authoring systems, such as Genis, and most of the authoring guidelines promote the use of an instructional algorithm that had been shown by research to have some merit. This is the tutorial algorithm in figure 1.

In this algorithm, the program presents new text information, asks about the information, judges the response and, if the learner was correct, presents new information. If the learner was wrong, it generally presents the learner with the same questions again with optional help or remediation available.

The algorithm in figure 2 (based on the theory of Robert Gagne[3]) goes beyond the simple specifications of presenting information, asking questions and providing feedback, even though fundamentally the same computer capabilities are utilized in both algorithms. That is, the functions of the computer are: 1) presenting text (one form of stimulus presentation), 2) asking questions, and 3) judging response.

These functions can be structured to follow a theoretically sound model such as figure 2 which is based on Gagne's theory.

One of the first steps in designing instructional materials is to categorize the types of learning outcome. These have been described by Gagne as follows: 1) verbal information, 2) intellectual skills (consisting of discrimination, concrete concept, defined concept, rule and problem solving), 3) cognitive strategies, 4) motor skills, and 5) attitudes.

Once the learning outcome of a lesson has been categorized the next step is to proceed with a sequence of steps that teach. By teaching is meant the presentation of a series of displays that support or enhance learning. This support occurs through events of instruction. In any complete act of learning there must be nine events of instruction. (See figure 2.)

Depending upon the learning objectives and the intended learners, the order of specific displays to represent each of the nine events can be varied. Sometimes one or another of the events is not included because it is evident to the designer that either the learner and/or the task make it unnecessary.

The nine events require different specific forms depending on the category of learning outcome. For instance guiding learning requires a quite different form of organization for the learning of verbal information than it does for the learning of a defined concept or rule.

INSTRUCTIONAL COMPUTING STRATEGIES

Strategies for Instructional Computing can be defined in six major areas: 1) tutorial, 2) drill and practice, 3) simulation, 4) games, 5) problem solving, and 6) testing. Each strategy may be represented by an instructional algorithm.

1. Tutorial

The tutorial contains the presentation of information to the learner, asking questions about the information and giving feedback as to the correctness of the response.

The "tutorial dialog" can assist the user in locating the specific cause of error(s), provide hints, or if needed, branch to a separate section for detailed review. The inportant distinction from other instructional computing uses is that the computer program acts to a limited degree as a surrogate tutor for the student. It is this very nature that makes valid programs of this type the most sophisticated, but most difficult to design and develop. It has been estimated that 50-200 person-hours are required for each student-hour at a computer terminal.

There are two general types of tutorial programs, linear and branched. The linear program gives the same information to all learners and asks all learners the same questions. The algorithm for a linear program is shown in figure 3.

The branched tutorial program gives the learner different text displays depending upon how he or she answered the questions asked in the program. Thus, all learners do not get the same text displays or questions. The branch algorithm might be diagrammed as shown in figure 4.

The branched type program gives more possibilities for remediation within a single program, but both linear and branched programs are subject to the same rule of design. Branched programs are more complex than linear programs, and therefore more difficult to design and program. This is why so many tutorial programs end up being linear.

2. Drill and Practice

The drill and practice algorithms assumes the user has already learned the content and that retention of the content will be facilitated by repeated practice. This algorithm lacks the "presentation of the stimulus" and "learning guidance" often found in the tutorial lesson. Students can learn new skills from drill and practice lesson, however, the learning strategy is trial and error. It is in general inefficient and is outlined in figure 5.

Drill and practice programs are second only in use to problem-solving applications in instructional computing. This technique also has wide application; any area in which certain fundamental concepts require practice for mastery.

This could be multiplication tables, chemical nomenclature, Latin-English word root translations, state capitols, etc.

Drill and practice programs are generally very straightforward. An introduction usually including examples is given; drill questions are presented (either linearly or by random selection); answers are entered and checked for accuracy; appropriate feedback is given; the next question is asked and so on. At the end of the program, some form of performance report is given and, perhaps, recorded.

3. Simulation

The simulation program can be described by the algorithm in figure 6. This type of program is often used to teach the relationship between variables by the "discovery" method. Typically, the events of instruction are to present the stimulus (information about a system). The learner is then asked how he/she might change the system (elicit responses). Feedback with regard to the effect of his/her information on the system is provided and the cycle is repeated. There is no formalized "learning guidance" and the learner must deduce the relationships among elements of the system by analyzing his/her inputs and their results.

Usually, simulation applications in instructional computing are used when a given concept is important from a pedagogical process, but one or more of the following apply:

a) there are time and/or space and/or equipment limitations;
b) the real process might place the student in a perilous situation; and
c) review and/or practice would be beneficial prior to performing the actual experiment of process.

Simulations in instructional computing are based upon models, most of which are mathematical in origin. In general, these programs allow a user to manipulate parameters and perhaps "discover" the effect of those manipulation. Popular examples include: 1) Population dynamics (what happens to the population over a certain span of years if both the birth and death rates decrease and the female/male ratio increases?, etc.); 2) Environmental "Studies" (what happens to the water oxygen if untreated raw sewage is dumped into a slow moving stream? A fast-moving river? What is the effect of performing primary treatment? Secondary treatment? How does temperature effect the results above?).

4. Games

The instructional game (see figure 7) is often claimed to be appropriate for sharpening problem solving skills, but the particular skills often go unidentified. The algorithm contains a stimulus presentation (like the simulation) and the elicitation of a response from the learner. However, the stimulus is then changed by an algorithm unknown to the user. In most instructional games the student is trying to beat the computer. The object of the games becomes to win (not necessarily to learn anything).

5. Problem Solving

The heaviest use of instructional computing to date is that of problem-solving--that is, writing computer programs to solve specific discipline-oriented problems. This particular application, for all practical purposes, has no limits! It could be finding the roots of a quadratic equation in mathematics, calculating lunar orbits in physics, gas law problems in chemistry, analyzing voting behavior in sociology, determining circulation trends in library science, etc.

6. Testing

Testing is an application similar to drill and practice, with the exception that no "assistance" is provided. A question is asked, user response is entered, and at some point, the user's performance is indicated. Another common example is the generation of examinations in the form of computer printout. Questions for a "unit" are selected from a "data bank," often in a random mode to make each exam unique. Answer keys are also generated or in some instances, the answers are entered on mark-sense sheets for scoring by the computer using an optical mark sense reader.

INTERACTIVE STYLE

The interactive style of your lesson is critical to its effectiveness as a teacher. Through your interactive style you can motivate the students and effect learning if your program generates all or some of the processes of internal learning. These are alertness, expectancy, retrieval, selective perception, semantic encoding, reinforcement and generalizing.[4]

Attention to the following points[5] will assist you in developing an interactive style.

1) Arouse interest and motivate the user.
2) Use variety in the presentation. Be careful, however, that variety does not become inconsistency, or you may confuse the user. Add variety through the use of graphics, sound, and amination if doing so supports the purpose of the program.
3) Involve the user through a degree of interaction and personalization.
4) Be careful of using sound in programs that may be used in a classroom setting where sound can disturb those not involved with the program.
5) The effective use of color can enhance the display, but try to avoid absolute dependence on color to accomplish this goals of the program.
6) A program should be easy to use. The topic of the program, not the mechanics of its use, should challenge the user.
7) Avoid system errors. Interaction between the user and the program should not result in an error that terminates the program. These potential errors can be minimized by providing response options to the user that (a) avoid excessive typing and (b) avoid confusion as to the response required.
8) Use menu selection as much as possible when presenting options to the user.
9) Avoid scrolling information by allowing the user to control movement from page to page.
10) Do not overcrowd the screen with information. Keep each screen centered on one point.
11) Do not mix a large amount of text and graphics.
12) Display a small amount of text at a time. Double space text material so it is easy to read.
13) Keep main ideas together, and highlight major points by sectioning them with lines or using reverse video.
14) Put yourself in the role of the learner.

In summary, theory-based design should consider the types of learning, learner characteristics, and the strategy in which the instructional computer program will be used.

I have covered a description of the type of content that would be included in each of the instructional events for verbal information, defined concept, rule using and problem solving learning (see figure 2).

Instructional Design

One of the first problems facing the author is to classify the objectives of the lesson so that the proper content for each of the events can be written.

Next, the author must determine the relationships among the objectives for the purpose of sequencing them so that prerequisite skills are taught first.

Then, utilizing the information model of Gagne, the author would produce the elements that present the events of instruction.

Instructional Strategy

With careful analysis of the learner and the objectives you determine the instructional strategy (i.e., simulation or tutorial).

The initial analysis and design of the program determines both the ease of programming and the quality of the product. Most successful programming techniques make use of structured programming.[6]

Interactive Style

You then design the interactive interface or the screen presentations. This aspect of the lesson defines the learner (student) to teacher (computer) interaction. Detailed planning is essential to high quality software.

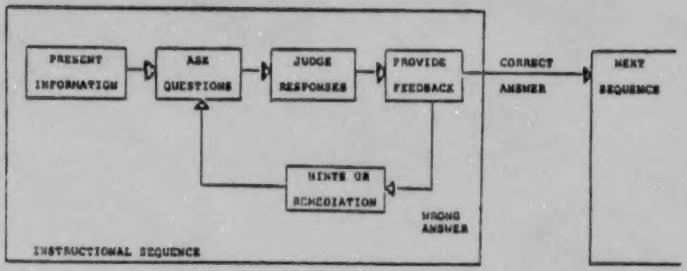


FIGURE 1. TUTORIAL ALGORITHM

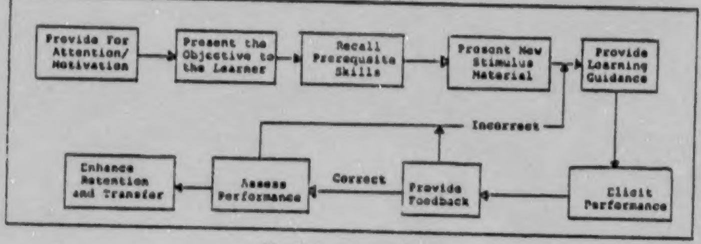

Provide For Attention/ Motivation
Present the Objective to the Learner
Recall Prerequisite Skills
Present New Stimulus Material
Provide Learning Guidance
Incorrect
Enhance Retention and Transfer
Assess Performance
Correct
Provide Feedback
Elicit Performance

Figure 2. Gagne algorithm

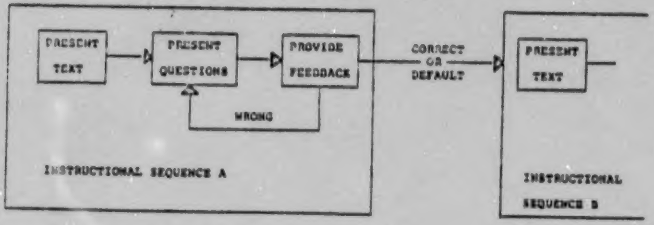


FIGURE 3. TUTORIAL LINEAR PROGRAM

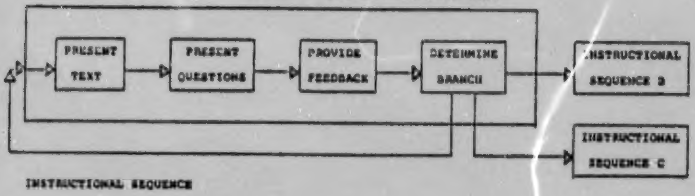


FIGURE 4. TUTORIAL BRANCHED PROGRAM

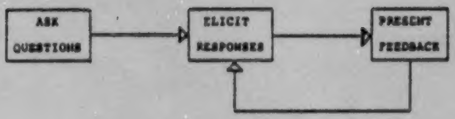


FIGURE 5. DRILL AND PRACTICE ALGORITHM

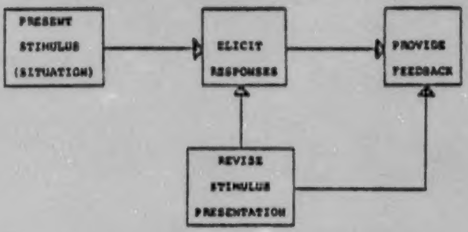


FIGURE 6. SIMULATION ALGORITHM

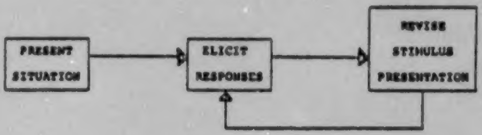


FIGURE 7. GAME ALGORITHM

[1] Two authoring guides, CREATE, 1978 by Control Data Corporation and CONDUIT's Authors Guide Design, Development, Style, Packaging, Review, 1978 at the University of Iowa offer guidelines in designing the lesson.

[2] "A Comparitive Analysis of Models of Instructional Design," Journal of Instructional Development, 1980, 3(4), 2-16.

[3] Gagne, R.M., and Briggs, L.J., Principles of Instructional Design, New York: Holt, Rinehart & Winston, 1979.

[4] Gagne, R.M., Wagen, W., and Rojas, A., "Planning and Authoring Computer-Assisted Instruction Lesson," Educational Technology, 1981, 9, 19-26.

[5] Hartman, Ken, "Authoring Considerations in Writing Instructional Computer Programs," The Computing Teacher, September 1982, 27-29.

[6] "Structured Programming for Personal Computing," Creative Computing, March/April 1978, is a good introduction for the novice.

WHY CONDUIT-
FROM THE POINT OF VIEW
OF A UNIVERSITY

Jo Ann Baughman
Oregon State University
Corvallis, Oregon 97331

ABSTRACT

Computing is a powerful technology that affects all aspects of everyday life. Because computing deals with processing information it has the potential to enrich and reshape education in all academic fields.

Today instructional computing falls far below its potential, leaving a rich educational resource virtually untapped. The reasons for the lag in computer use are numerous, but recent studies of the barriers to widespread use of computer-assisted instructional have pointed out that with the current dramatic reduction in equipment cost, the single most important barrier is the lack of quality curriculum materials.

The Educational and Research Services group at the OSU Computer Center has used CONDUIT to enhance the software available to the users.

1. INTRODUCTION

Few technological developments have stirred as much fantasy and despair as computing; and rightly so. Computing is an intellectual tool which helps us manipulate information. It allows us to deal with organized complexity. While some are busy debating whether they are a modern day equivalent of the steam engine or the printing press, computers are silently entering retail stores, banks, communication systems, industrial plants, saw mills, and the home. Computing is becoming the largest industry in the world.

It is astonishing that computers do not play a larger role in education than they do. They are widely used for administrative tasks and for research. However, instructional computing falls far below its potential. The reasons for the lag in instructional computer use are numerous, but recent studies of the barriers to wide-spread use of computer-assisted instruction have indicated that with the current dramatic reduction in equipment cost the single most important barrier is the lack of quality curriculum materials. This is closely followed by the lack of trained teachers as a second critical barrier to instructional computing.

If the acceptance of instructional computing is to be speeded up, barriers to its use must be identified and systematically removed. Computing needs to become easier and more economical to use. Quality instructional computer materials must be readily available. CONDUIT is addressing these problems.

2. CONDUIT

CONDUIT was established in December 1971 to investigate the immobility of computer-based instructional materials in undergraduate education. The study brought together five major regional computer networks, each with unique resources and configurations serving over 300,000 students. These were Oregon State University, North Carolina Educational Computing Services, Dartmouth, and the Universities of Iowa and Texas.

CONDUIT is a nonprofit organization whose purpose is to promote the use of instructional computing today. CONDUIT seeks to accomplish this purpose by providing instructors with ideas, curriculum materials, and information about instructional computing, and by encouraging and assisting instructors and development teams to produce computer based learning materials that will be useful to educators.

Computer based instructional materials do exist, but they are often incomplete, of poor quality, or unsuited for use at more than a single site. CONDUIT believes that many of the barriers can be reduced once the efficacy of widespread distribution is demonstrated. Some of the direct benefits of widespread distribution include: increased author recognition and reward, lowered development costs resulting from shared expenses, improved quality of materials and techniques, reduced duplication of effort by building upon the work of others, and increased use of standard techniques, thus reducing technical incompatabilities.

Above all, CONDUIT seeks to distribute quality materials. Their aim is to provide instructors with a single, reliable source for ideas and software, and to assist authors by performing the burdensome task of distributing the materials.

The cornerstone of their activities is the review process. The first step in the process is peer review, where educators with experience in the content area and in using computer based materials examine and use the materials in the classroom. Reviewers report their findings using the CONDUIT Review Rating Form, especially designed to evaluate computer based materials. From these reviews, CONDUIT is able to determine the material's conceptual validity, instructional usefulness, and overall quality. Leading educators in several disciplines coordinate the reviews and advise CONDUIT concerning the educational elements of each unit.

Once a unit has passed the substantive review, the CONDUIT staff then evaluates each program for accuracy and prepares the program so that it can be easily installed and used on a variety of computers (both large and small, batch and interactive, Basic and Fortran).

The materials offer a variety of approaches to instructional computing. Several packages are computer simulations of situations that are impossible or too expensive to reproduce in the undergraduate laboratory. Other packages help teach quantitative techniques in such courses as linear algebra, calculus, operations management, and quantum mechanics. Some of our materials provide problem-solving tools or data for student research. We also distribute packages which tutor students in key concepts or provide drill and practice in basic skills.

CONDUIT has gained an international reputation for its distribution of materials and has been referred to as the best source of instructional computer materials. If you order their materials and find them unsuitable, you may return them. Their return rate is less than two percent.

CONDUIT is continually studying ways to make instructional computing more effective. Their current activities include research and development of aids for authors, programming standards, transfer guidelines, and evaluation tools.

3. CONDUIT AT OSU

CONDUIT at Oregon State University is part of the Research and Instructional Computing Services of the Computer Center.

The CONDUIT Coordinator at OSU is also responsible for Faculty Workshops, Research and Instructional use of computers on campus, and faculty support in the selection and use of software on the Research and Instructional Program Libraries.

CONDUIT provides Oregon State University Faculty, through the Computer Center, with quality computer-based instructional materials that are easy to maintain and support. Since 1971, our programmers have become experienced in Transfer problems.

Over a two year period, data has been kept relating to programming efforts in the installation, maintenance, and support of programs in Educational Computing Services. Table I details this information. CONDUIT materials are roughly one-tenth as expensive to the University as locally developed programs. CONDUIT materials are one-sixth as costly to the University as programs transported from non-CONDUIT universities or Research Laboratories. CONDUIT has proven to be an excellent source of instructional computer materials. CONDUIT[1] has also prepared a series of reports and guidelines for importers and exporters of computer-based instructional materials.

TABLE I

PROGRAMS	LINES/ HOUR	@ $20/HR $/LINE
CONDUIT	172	$ .12/ln
Other Sites	25	$ .80/ln
Developed at OSU	16	$1.25/ln
3300 To Cyber Conversion	59	$ .35/ln

4. INSTRUCTIONAL COMPUTING AT OSU

At Oregon State University Instructional Computing constitutes 16 percent of the total usage of the Center. This usage is 58.9% of the accepted standard established by the President's Science Advisory Committee on Computers in Higher Education (1971).

Current budget levels restrict instructional usage, with the result that departmental requests must be cut by as much as 25% each term. Expenditures total less than $15 per FTE student, which translates to less than 60% of the most commonly applied standard for student support.

Instructional usage at OSU is, however, distributed across most disciplines, with over 500 courses per year employing the computer to some extent. Over one-half the usage is concentrated in Engineering and Computer Science.

Almost 9,000 students per term are enrolled in courses which use the computer. The area of programming skill development in Computer Science accounts for 50% of the use. Problem solving accounts for 50% of the instructional computer usage. This includes the use of canned programs, simulations, and CAI. For example, students in Anthropology study the characteristics of World Cultures through an interactive graphics program. CONDUIT contributes to this instructional usage through economical programs.

---

[1] The University of Iowa, CONDUIT, 100 LCM, Iowa City, IA 52242.

Despite this, instructional computing is the most visible problem, and perhaps the most critical as well. The identified need for student computing has not been fully met in any of the last five years because of budget limitations. Even so, the level of support has increased 82% in the last two years.

The rapid growth of the Computer Science Department has been a major factor in the failure to satisfy needs. In 1976-77, Computer Science used 26% of the instructional computing budget; in 1980-81, that figure had grown to 50%. In the last two years, the cost of serving Computer Science students has increased 159%. And yet, Computer Science has probably compromised more than any other department in trying to keep expenditures within budget.

TABLE II

DISTRIBUTION OF
INSTRUCTIONAL COMPUTING

DISCIPLINE	1978-79	1980-81
Computer Science	38%	50%
Science (except CS)	17%	9%
Engineering	21%	21%
Business	8%	7%
Liberal Arts	6%	5%
Other	10%	8%
TOTAL	$236,377	$428,732

Oregon State University, however, is not at ground zero in the area of instructional computing with 9,000 students out of a total enrollment of 16,000 using the computer each term. OSU currently supports a network of over 500 terminals which have access to either Control Data Corporation (CDC) Cyber 170/720 and a Honeywell Information Systems (HIS) level 66/DPS model 440. The latter machine is primarily devoted to student use. The Cyber is available to all categories of users. There are also over 150 microcomputers located in departments primarily for research use but some are also used by students.

Access to quality software through CONDUIT has contributed to this growth in the instructional use of the computer.

WHY TEACH ABOUT COMPUTERS?

Phillip J. Berrie, Ph.D.
Educational Services Division
Heartland Area Education Agency 11
1932 S.W. Third Street
Ankeny, Iowa 50021

The phenomenal growth of microcomputer availability for K-12 classrooms has posed educators with the problem of determining the role of the computer as an integral part of a student's education. To assist in seeking solutions to that problem, the Iowa AEDS Chapter sponsored several Computer Literacy Workshops involving teams from school districts throughout the state. This paper shares the highly successful techniques used by the workshop participants in developing computer literacy rationales, the beginnings of K-12 computer literacy scope and sequences, and specific computer literacy implementation plans.

In conjunction with the November, 1982 Iowa AEDS Annual Conference, a pre-conference workshop on computer literacy drew over 130 participants representing thirty-eight school districts throughout the state. Since then, three more identical workshops were held bringing in another 286 educators from ninety-six districts. The popularity of the workshops came about for two reasons: 1) educators are beginning to realize that computer literacy is another educational "basic", and 2) the format of the workshop produced serious planning and immediate results.

School districts were invited to send a three member team composed of at least one elementary teacher, one secondary teacher and one administrator. The advertised purposes of the one-day meeting were to:

1. share ideas and concepts with other districts.
2. clarify and formulate their district's definition of computer literacy.
3. organize and develop the beginnings of a K-12 computer literacy scope and sequence.
4. identify steps to implement a computer literacy plan for their district.

To accomplish such lofty goals it was necessary to choose between bringing in renowned experts to "tell it like it is" or allowing the district personnel to decide what was appropriate in their own situation. The latter approach was used by the facilitator in order to have each district be able to have "ownership" of their computer literacy plan.

After the normal introductory remarks to set the stage for the day, the participants rearranged themselves into job-alike groups of five, e.g. five elementary teachers each from a different school district. Their task was to respond to the question "Why teach about computers?" using a highly structured process called the Nominal Group Technique (NGT).

The NGT involves several steps: 1) individual silent generation of ideas in writing, 2) round-robin recording of ideas with no discussion, 3) serial discussion and clarification of ideas, 4) preliminary vote on item importance. Thus, by the end of the first session, each group of five educators came to some consensus on why something should be taught about computers.

The participants were then asked to regroup back into their district teams bringing with them the ideas discussed and evaluated within their respective job-alike groups. Their task became one of writing a rationale for a computer literacy program for their district.

Within an hour, each team had thoroughly discussed their points of agreement and disagreement and had written a rationale that they could take back to their schools. Many felt that with a little polish, they could present it to the school board. A representative example is found below.

Since computers are just another in a long progression of tools that have been developed by humans to ease their burdens, we propose that the __________ School District should introduce computer literacy as an interdisciplinary course. Students need to learn about computers because of social, economic, academic and personal implications.

Social implications should address the history, current uses and future trends. To be responsible citizens, students need to develop values of how data should be used and to realize the possibilities for abuse. It should not be an elitist tool, but computer technology should be guided by a large base of citizens as a social benefit. They need to know the value of accuracy; garbage in, garbage out.

Economically speaking, students need to be aware of not only career opportunities with computers but also that being able to use a computer will become a necessary basic skill to survive in the job market.

Academically, students should understand the need to develop problem-solving techniques and systematic thinking. They need to know that information is available through networking and data base resources.

Students will benefit personally by replacing fear of the computer with confidence and the power to control it. Creative satisfaction of interaction with the computer would be a personal benefit. They need to be able to evaluate technological change and the capabilities and limitations of the computer.

By being exposed to computer literacy in the public schools, students will be better prepared for post high school education and vocations.

After a break for lunch, the group reassembled and heard a general presentation concerning definitions of computer literacy as found in the literature. It was pointed out that three components are usually expressed, 1) awareness of computers, 2) passive use, and 3) active use. The awareness component could consist of learning about the uses and abuses, history, personal and social aspects, attitudes, and, perhaps, hardware/software principles. Passive usage of the computer involves, but is not limited to, being able to get information into and out of the computer, using packaged programs, and knowing how to evaluate software. Active usage assumes a facility with algorithms, programming and computer languages.

Immediately, after a short question and answer period the participants once again formed job-alike groups of five. Using the NGT process, they were to investigate the question "What should be taught about computers at the elementary (secondary) grade level." Resource materials which included scope and sequence charts and journal articles were available for examination. Once again the topic was thoroughly discussed and set the stage for the next task at hand.

Regrouping into district teams, participants were to now examine what was currently being done within their district and what needed to be done to live up to the rationale that they had previously developed. Using a blank scope and sequence chart, they attempted to broadly sketch which concepts should or could be presented at which level. Within less than two hours, many teams had come to some agreement and had the makings of their own computer literacy curriculum plan started. Other teams had to spend more time ironing out philosophical differences but at least they were able to recognize where they differed.

The last task of the day may have been, in many respects, the most important one. Each district team was asked to identify tasks to be done in the future in order to continue the process started during the workshop. As the tasks were listed, the person/group responsible for the task was identified, as was the completion date.

At the end of the day there were many exhausted individuals due to the intense discussion and effort demanded of each and every one of them. An examination of several of the evaluation comments does a more than adequate job of stating that the effort was worth it.

"It is good to go beyond theory and looking at other districts' programs. Thanks for a very pragmatic experience. Today has saved us weeks (possibly months) of work. Your means of structuring our work is a stroke of organizational genius. --Thanks--I'll use these methods again in other settings."

"One of the better conferences I have attended in the last 6 years. Being an administrator and not knowing a great deal about computers, talking to other people who have knowledge in the area was very helpful. Enjoyed the discussion sessions."

"This has been a rewarding experience, mainly for one reason...that is, this has forced us to make a beginning, with advice, guidance and ideas from several other sources. We will continue this and see it through to its implementation in our district. Your presentation and structure were very good, and appropriate. Thanks."

"My level of understanding regarding Computer literacy was minimal prior to this meeting. After attending, I feel I have a much better understanding through the Job-A-Like discussion groups, team meetings--activities such as developing rationale, scope and sequence, outline of next steps. I now feel a higher level of understanding and the ability to assist with the computer planning for the district. To sum up, very worthwhile and practical."

"I feel that a lot was accomplished today. It was very beneficial having an administrator as part of the team. I got a lot of information from teachers from other schools. The day was organized well. It ran smoothly with good input. I succeeded in realizing that there is a tremendous amount of work still ahead. We need more meetings of this kind."

STRUCTURED SYSTEMS TECHNIQUES USED IN LONG RANGE PLANNING FOR ADMINISTRATIVE EDUCATIONAL SYSTEMS

Edwin E. Blanks
Virginia Commonwealth University
Richmond, Virginia 23284

Virginia Commonwealth University developed a comprehensive five year administrative systems plan during the winter months of 1981. Prior to this time, the increasing demand for data processing support systems became so great that the efforts to address these prodigious needs continued to lag behind user requirements. The administrative systems analysis and design needs which are suited for an organizational environment with our complexities are certainly easier to state than to plan for and establish priorities.

Traditionally the administrative support systems at the university had been developed along functional lines while ignoring both management reporting needs and data flow among departments. The systems development life cycle methodology (SDM-70) was being used to insure standards, documentation, user deliverables, and technical design decisions. It addressed specific department factors but constrained the potential interface options.

Prior to beginning the planning study, a series of meetings were held with senior university officials to ascertain management support. Typically upper level management requires cross-functional summaries, special requests, and answers to what-if type questions. It is also important to recognize the varying characteristics of information as needed by different departments and management levels. Several of our vice presidents suggested establishing three distinct planning and control levels at the university. These are:

- Strategic Planning: The process of deciding the organizational objectives, the required resources, and the necessary policies to govern the acquisition and use of these resources;
- Management Control: The process by which managers assure resources are used efficiently; and
- Operational Control: The process of assuring that specific tasks are carried out effectively.

It would seem impractical to construct a single system to accommodate all of these activities or management levels, and it would be erroneous to associate any one type of information requirement solely with one management level.

The development of the plan began with a detailed study. Four university analysts were assigned the responsibility to conduct the study, analyze the findings, and prepare the plan. To assist the efforts of these analysts, a consultant from one of the big eight accounting firms was employed. The engagement stipulated management consulting services in the following four areas:

- Development of planning methodology using structured techniques
- Project guidance and monitoring
- Final plan evaluation
- Training the analysts to use the structured tools

The informational needs identified in the approximately 40 interviews with administrators from a cross section of departments were summarized, organized, and grouped into five administrative process areas. These are:

- Financial
- Human Resources
- Academic
- Student
- Facilities and Inventories

Informational needs associated with each process area are identified and itemized in the plan.

The design section of the plan depicts a structured format consisting of a catalog of the inputs, outputs, data stores, and informational processes. The design objectives for each of the identified systems conclude with a data flow diagram (DFD) which graphically represents the interconnections among the system's major informational processes. As one example, see the Financial Billing and Collection Data Flow Diagram. These systems are shown at the logical level using the symbols below to construct the Data Flow Diagram.

DATA FLOW DIAGRAM (DFD)

Double Square		Source or destination of data
Arrow	→	Flow of Data
Rounded Rectangle		Process which transforms flow of data
Open Rectangle		Data Store

Each process box can be exploded into a lower-level or more detailed DFD.

In summary, the analysts felt this technique was fairly easy to learn and impart to the users. Their most difficult task was ostensibly in starting to think at the logical level rather than in terms of physical implementation.

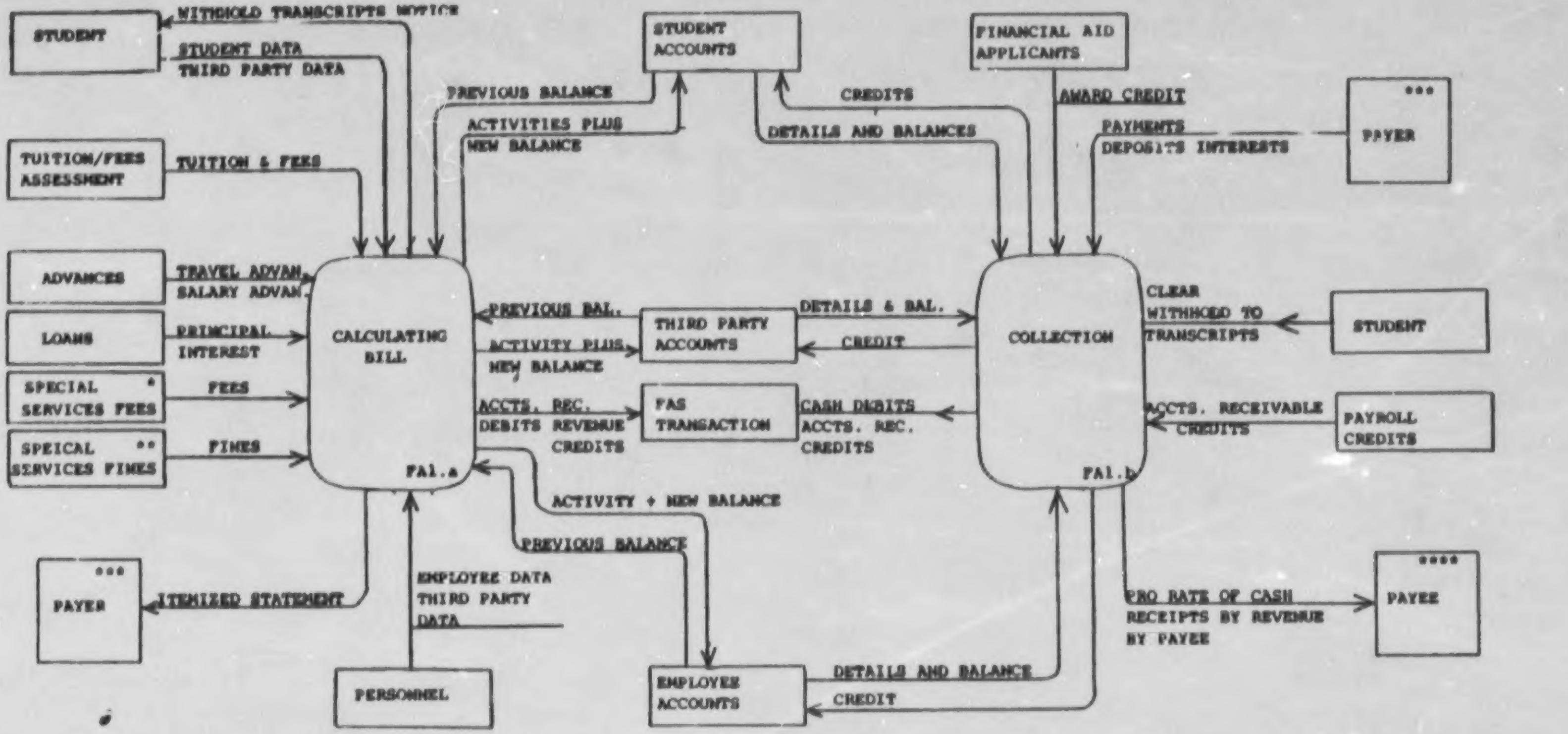
During briefing with users the following points from the plan were successfully stressed:

- Top Down Structured Development
- Reassurance that this new technique replaces the existing approach rather than imposing yet more effort on the part of the user
- Explanation of the Data Flow Diagram approach
- Efficiency of using an automated data dictionary to hold the glossary of items resolves those cases where people call the same things by different names.

BIBLIOGRAPHY

Datamanager Implementation Plan and Management Report, Management and Systems Programming Limited, Lexington, Mass., 1979.

Gane, C. and Sarson, T., Structured Systems Analysis: Tools and Techniques, Prentice Hall, Englewood Cliffs, N. J., 1979.



* HOUSING
BOARD
TRANSCRIPTS
GRADUATION
ID CARD REPLACEMENT
PARKING PERMITS
LOCKERS
MCV CHARGES
TUITION WAIVERS

** LIBRARY FINES
PARKING TICKETS

*** STUDENT
EMPLOYEE
THIRD PARTY

**** REGISTRAR
LIBRARY
PARKING
PHYSICAL ED
MCV
ACCOUNTING

BILLING AND COLLECTION

FA1

MODERN APPROACHES TO LEARNING TO PROGRAM

Alfred Bork
Educational Technology Center
University of California
Irvine, California 92717

March 9, 1983

The following comments form an extended abstract of the talk to be given at the meeting. The form followed will be to make a statement in each case, and then give some supporting information about that statement. In the talk itself examples of each case will be used, and more details will be presented. A more complete discussion of these issues will be presented in a forthcoming book about computers in education, Personal Computers in Learning--An Introduction.

Computer literacy is not programming.

The content of a computer literacy course is not the subject of these comments. There is tremendous vagueness about that issue, with the term "computer literacy" used in a variety of ways. The teacher should not confuse computer literacy and programming.

Programming is very different from what it was fifteen years ago.

Many junior and high school programming courses are not teaching programming as it is practiced today, but rather are teaching strategies that have long since been discredited for practical programming. It is extremely important to give students a modern point of view.

It is much more important to teach style than to teach grammar.

Many programming language courses seem to believe that the whole object is to teach people the grammar of the language. If one is to write programs today, the style of programming, usually called structured programming, is much more critical than the details of a particular language. Hence, any programming course should focus on the issues of style and structure, and these should receive greater attention than any issues of grammar. The features of modern structured programming are well known and will be discussed later.

The first habits in a particular area, such as programming are difficult to correct later.

If students are introduced to programming initially in such a way as to build up a set of bad habits associated with poor programming style, it is very difficult or even impossible to change these bad habits at a later stage. Hence, it is

The Center is also dependent on an group of extremely good undergraduate student coders. Current coders are:

Stephen Bartlett
Ben Boronow
Richard Clark
Elinor Colman
Colleen Kinney
Grace Lam
John McNelly
Alastair Milne
David Muhonen
Jeffrey Oswald
Greg Roberts
Naomi Salvador
Jina Tan
Patrick Young

In addition to its research and development activities the group gives workshops and consults for school districts, universities, and commercial concerns. We will be happy to discuss this further with interested parties.

RESEARCH AND DEVELOPMENT AT THE EDUCATIONAL TECHNOLOGY CENTER

Alfred Bork
Educational Technology Center
University of California
Irvine, California 92717
(714) 833-6911

The Educational Technology Center is a research group at the University of California concerned primarily with computer based learning. We carry on a wide range of activities in research and development. The following list identifies our current projects. Each will be discussed in more detail at the AEDS convention.

Research Projects

1) Screen design and readability of computer materials (NSF)
2) Groups and computer based learning (NSF)
3) Teacher characteristics and successful classroom use of computers - NIE with Rand Corporation)

Development Projects

1) Scientific literacy in public environments, schools, and homes (FIPSE)
2) Formal reasoning capabilities in early adolescence (NSF)
3) Computer literacy for teachers (NSF)
4) Problem solving in real-life situations in junior high school, with associated video (AIT)
5) Retaining students in beginning physics (FIPSE)
6) Translation and marketing of physics quizzes
7) Commercial training material

The Educational Technology Center is a relatively small group, entirely dependent on outside funding for particular projects. Current senior staff members of the group are

Alfred Bork
David Trowbridge
Werner Feibel
Ruth Von Blum
Christopher Wells

The supporting administrative staff is

Cynthia Powell
Mary Doi
Dorothy Gallatin

particularly critical that correct style e learned in the earliest exposure to programming. Evidence at the college level indicates that if students have not seen correct programming style, they perhaps will not be able to continue with computers at the college level.

Encourage students to work in groups.

Most modern programming cannot be done by an individual. At an early stage people must begin to understand that they have to work with others on a large project. So part of modern programming practice is to develop group cooperation and group style.

Reading books and listening to lectures are not the best way to learn programming.

Programming is an active activity. Students must be active participants in the process if they are to program successfully. Hence, methods of programming which encourage student interaction, with the computer and with each other (peer learning activities), are important to emphasize in beginning programming courses.

Program examples are an important way to learn to program.

The student should see many examples, developed in modern structured form and should discuss these examples. Using examples particularly relevant to student interests is useful. Thus, if programming is being taught within a mathematics course, some real subject matter areas are available with natural examples to use.

Don't teach or learn BASIC as a first language.

While it is barely possible that BASIC could be learned in a way that is consistent with modern programming, it seems extremely unlikely that this will happen. Even in universities, only a very small percentage of courses in BASIC are taught so as to promote good habits. BASIC does not lend itself to many of the aspects of modern structured programming. I am not impressed by arguments that point out that BASIC is commonly available. So are candy bars, but we don't urge babies to subsist on a diet of candy bars. BASIC is the junk food of computer programming.

Use a modern structured programming language.

In the earliest exposures to programming it is important to work with a structured approach, a language which encourages structured use. This implies that it must have reasonable procedures, must allow reasonable variable names, must have decent control structures, and must allow careful specification of data types. "Allow" is probably not the right word; "insist pon" is a better term. Languages such as Pascal are the most likely candidates.

Investigate the Computer Power course.

The Computer Power course, available from the Gregg Division of McGraw-Hill, seems to be one of the best possibilities for high school programming courses at the present time. It is based on a subset of Pascal initially, and then graduates to full Pascal. The problem with teaching programming in schools is the problem of finding decent curriculum material. Very little of this exists at the present time, especially material that follows modern structured programming practices.

Be clear about WHY you are teaching programming

This is the most difficult piece of advice. Different reasons for teaching programming might lead to very different types of courses. For example, if you are training people who will go on to college in computer science, it is particularly important to observe what the standards are. The recent Advanced Placement exam in computing, for example, furnishes a reasonable standard to follow.

EQUIPPING TEACHERS FOR THE UNFINISHED REVOLUTION: SOME UNIVERSITY COURSES ON COMPUTER EDUCATION

ANN BREBNER
University of Calgary
Calgary, Alberta, Canada

ABSTRACT

An account is given of the development and rationale of a set of university in-service courses on the topic of computers in education. Several such twenty-hour courses are described concerned with computer literacy and a variety of more specialized topics. Ways are discussed in which the content for these and other courses may be determined.

THE NEED FOR COURSES IN COMPUTING

Within the last few years it has become quite commonplace to find the term "revolution" used to describe the rapid rate at which computer technology is advancing, and in 1983 all indications are that both hardware and applications will continue to be developed at the same pace, or even faster, for many years to come. As a result, the computer revolution will have an increasingly significant impact on many aspects of society as we know it today.

In education the speed of the developments has already created a series of situations which have no parallel in the past. For example, at no previous time has it been suddenly necessary to devise curricula for a topic at all levels from elementary school, through high school, to college and university, and from introductory courses to more advanced ones.

Also, such curricula are often being developed and implemented by teachers who have not themselves had sufficient time to obtain an in-depth knowledge and experience in the area. Further, teachers face classes with a wide range of computer knowledge among the students, from those who know little to students more competent in the area than their teacher. Again, this situation is one which has not occurred before, and is a direct result of the speed of the computer revolution. Most practising teachers today did not take computer related courses during their university training period because the need to do so was not evident at that time.

These problems are further compounded by the fact that those with any skills in the area often find heavy demands on their time, and have little left to further their own knowledge.

To meet the sudden demand placed upon the education community for a computer literate population, a key factor is to provide education first for the teachers.

IN-SERVICE COURSES

To address this need, in-service courses were started in the Faculty of Education at the University of Calgary in the Spring of 1980. There is no doubt that teachers perceive them as fulfilling a necessary function as they have expanded to the point where 333 students were enrolled in the Fall of 1982, and only enrollment limits imposed due to lack of hardware prevented this number from being higher. The courses are quarter course equivalents, that is they consist of 20 hours of instruction.

The initial course is an introduction for teachers with no knowledge of computing. When it was first given, the classic mistake was made of trying to pack too much material into the course. An overview of how computers are used was given by a series of guest lecturers, each addressing their special area of interest. These ran in parallel with lectures by the main course instructor who covered the topics of computer development, the architecture of a microcomputer, and an introduction to the BASIC language. As a result of experience and student feedback, the time spent on the overview was greatly reduced and is allotted instead to the teaching of BASIC. Hence there is now a strong practial aspect to the course and students spend about 50 percent of the time working at a computer. To date they have usually worked in pairs, mainly due to lack of hardware, but there are advantages to such an arrangement at this level. For beginners there are so many details to which they must attend, any one of which, if incorrect or overlooked, can prevent them from making any apparent progress. Working in pairs tends to alleviate this problem.

The hardware used is the same as that found at present in the local schools, and any programming texts required or recommended at this level relate exactly to the hardware, and to the version of the language implemented on this hardware. Small discrepancies of any kind, which are trivial to more experienced users, are very confusing to students at this stage of understanding, where many are still apprehensive about computers, and unsure of their ability to control them.

An introductory course, particularly one of short duration, is perhaps the most difficult course for which to determine the content. It should be a complete introduction for those who do not continue in the area, a sound foundation for those who plan to take further courses, and at the same time should contain a substantial amount of practical work. The chief difficulty is to decide what information to omit. It is obviously not possible to cover any topic in depth, but it is necessary to ensure some understanding of underlying concepts and principles.

More specifically the present content is as follows:

1. Rapid development of computers - very brief outline of their history.
2. Relevance of computers in education - computer literacy, programming, computer assisted instruction, computer managed instruction, and administrative uses.
3. Simple model of a computer - central processing unit/control unit, memory, input, output.
4. Digital computers - binary, distinction between program and data, bytes.
5. Microcomputers - microprocessor, RAM, ROM, EPROM, floppy and hard disks.
6. BASIC language
   - a) Immediate mode
     - output instruction for numbers and text - PRINT;
     - arithmetic operators and their hierarchy, simple expressions;
     - function of separators and terminators - comma, semicolon,CR;
     - assignment instruction - LET;
     - concept of a variable - address and contents;
     - rules for numeric and string variable names;
     - use of variables in expressions;
   - b) Program mode
     - line numbers and END instruction;
     - use of LET and PRINT in a program;
     - preparation of a floppy disk for saving a program;
     - use of the commands - NEW, RUN, LIST, SAVE, LOAD;
     - distinction between instructions and commands;
     - input instruction - INPUT;
     - unconditional branching instruction - GOTO;
     - concept of an infinite loop and how to terminate it;
     - conditional branching instruction - IF...THEN;
     - loops - FOR and NEXT instructions including use of STEP;
     - clearing the screen and cursor positioning.

It is not foreseen that any changes will be made to the content in the near future.

A second general in-service course on computer literacy, for which the course described above is a prerequisite, is also available. Again there is a substantial practical component, and again the content is divided between applications of computers and concepts in the BASIC language. There is more emphasis on CAI, and CMI is also discussed in greater detail. One reason for this is that CAI programs are being evaluated at present by the Provincial Department of Education, and some are likely to be available throughout the province in the near future.

The topics covered in the course include:

1. BASIC language
   revision of elementary programming concepts;
   steps in constructing a program - precise definition of problem, flow charts, testing, debugging methods;
   READ, DATA and RESTORE instructions;
   the use of a printer;
   concept of built-in functions - random number generation;
   concept of subroutines - GOSUB instruction, structured programming;
   introduction to use of color and graphics.
2. Computer assisted instruction
   demonstration of some available programs;
   evaluation of courseware and Provincial Clearing House.
3. Computer managed instruction
4. Brief introduction to software for word processing, data base management, and electronic spread sheets.

A variety of courses related to specific subject areas have also been offered at the in-service level. These have included computers in special education, computers in the basic subjects, computers in business education, computers in second language teaching, and computers in industrial arts. Due to the knowledge level of the student group, to date these courses have tended to follow a similar structure to the initial general course, and it has been possible to address the specific curriculum areas only in relation to applications and example programs.

Recently, because such packages are coming into widespread use, two new in-service courses have been set up, the first devoted entirely to word processing and the other to data base management systems. The first general computer literacy course is a pre-requisite for both courses. This is because the steps required to operate the packages are much more meaningful to those who have a basic understanding of the parts of a computer and how they interact, and who have at least a minimal experience of programming. Those without such knowledge do not see the need for such precise placing of each format character, and often lose their work because the reason for storing it on disk is not evident.

It is expected that there will be a continued high demand for in-service courses in the future by teachers wishing to keep pace with computer developments and applications. The content of the courses must evolve as teachers gain knowledge in the area; it is anticipated that enrollments in the two initial general courses will soon drop considerably and that the more subject specific courses will replace them. It is intended to continue to maintain a strong practical component in all these courses and for this purpose to use hardware that is, or will shortly be, in use in the schools.

Wherever possible courses will be scheduled over an eight week period with approximately 2-3 hours of instruction per week. After experience with a variety of time periods, including the packing of the course into one week, it appears that this is the optimal arrangement to maximize learning. With shorter periods students have difficulty absorbing what are radically new ideas and concepts for many of them, while with longer ones students tend to forget material between meetings. Naturally, courses are considerably more effective when hardware is available on which students can practise between classes.

It is also intended to maintain the present policy, whereby most of the instructors are practising teachers in the local schools who are using computers effectively in their own classrooms. These instructors are either former or present members of the strong graduate program which has been operating in the Faculty since the late 1960's.

GENERAL GUIDELINES FOR CURRICULA

Regardless of whether the computer will ultimately take its place as just another tool, with its applications taught within the different disciplines, over the next few years there is going to be a proliferation of courses related specifically to computing.

In the very near future most will be concerned with computer literacy, that is the body of knowledge required to function competently in a society in which the computer is becoming a ubiquitous tool. But consideration must also be given to the curricula of courses which lie beyond an initial introduction to the topic.

From within the revolution it is a difficult task to design the courses in such a way that they are both relevant today, and at the same time give information and teach skills which will remain useful, as the computer revolution continues.

However, there are perhaps some general guidelines which should be considered when deciding upon course content. First it is advisable to look back over the history of computing and determine those concepts which have remained constant over a long period. Some examples are the basic parts of a computer and their function, which could be said to have been conceived by Babbage in the early 1800's; use of binary numbers to represent instructions and data, which originated with von Neumann in 1950; and the concept of high level machine independent languages, suggested by Grace Hopper in the mid 1950's. An examination of these widely used general purpose high level languages developed since that time also reveals that there are many concepts common to most of them.

It is also necessary to look ahead and attempt to assess the significance of the leading edge of technology. For example, at the present time it appears likely that small relatively inexpensive machines capable of running the BASIC language only are likely to have a significant effect at least in the short term, while the impact of the fifth generation computer systems with emphasis on data flow architecture and AI type languages is as yet unknown.

In a topic where it is easy to become obsessed with current details and trivia, it is important to take this overall view. In so doing it becomes clear that emphasis should be placed on certain seemingly fundamental concepts of computing, and that hardware and software dependent topics should be avoided.

Since the computer is a tool, however, it is necessary that almost all courses contain a large practical component. This implies that students need to know machine and language specific details.

It follows that curriculum designers and instructors must keep a careful balance between these and the teaching of general concepts of computing. The apparent conflict is not as severe as it first appears and with care can be minimized. It is possible to structure the teaching in such a way that students can easily move from machine to machine and, knowing one high level language, can fairly readily change to programming in another.

It appears that only in this way can an element of stability be introduced into the curricula of computing courses, so that students can acquire information and skills which will remain useful to them for some time into the future.

This does not mean that content should not change. At this stage in the revolution the rapid rate of developments inevitably implies continuous updating of the content of any computer related course at any level from elementary school to university.

It does mean, however, that the latest hardware, software, courseware and support materials should be evaluated by those knowledgable in educational computing and introduced into the curricula only after very careful consideration. In this way some mistakes made in the past could be avoided, such as, for example, the introduction on a large scale of every fad in education regardless of whether there is proof of its effectiveness, and the reproduction on the microcomputer of errors made in computer science as the large mainframe computers evolved.

The unique circumstances of the computer revolution will inevitably produce many new challenges and problems both in the introduction of the technology into education and in courses which address the topic. It is essential to be on the right path in order to arrive at the "Frontiers of Educational Computing".

DIAGNOSIS SUBROUTINES IN DRILL AND PRACTICE PROGRAMS

George W. Bright
Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115

Diagnosis of students' skills and prescription of remediation activities seem to improve learning. Microcomputer drill and practice programs currently ignore this fact and typically allow students to practice whatever skills have been learned, even if the skills are incorrect or inefficient. Incorporating diagnosis subroutines into drill and practice programs might improve the effectiveness of these programs. Outlining the logic of these subroutines is one goal of this paper.

School instruction is typically paced for the group (class) rather than for individuals in the group, so there is often within a class both lost learning and incorrect learning. The former might cause gaps in performance; e.g., I don't know what to do!; while the latter may result in incorrect responses; e.g., $3x + 4 = 7x$. Diagnosis is a means for identifying both lost and incorrect learning. After a correct diagnosis has been made, remediation can be focused on precise difficulties and will hopefully be more effective than general remediation which is aimed at a broad range of possible difficulties.

Diagnosis is typically time consuming, both for the teacher and for the learner. If diagnoses could be made during the early learning of a skill or concept; for example, as part of drill and practice programs; students and teachers alike might save time and face less frustration, and more learning might occur.

Employing computer-controlled diagnosis as part of computer assisted instruction is possible (Travis, 1978; Woerner, 1980). Too, diagnosis does seem to improve achievement (Fisher, Berliner, Filby, Marliave, Cahen, & Dishaw, 1980; Okey, 1976). Sophisticated electronic devices projected for educational use (Evans, 1979) may in fact have to rely on diagnostic subroutines to monitor the development of concepts and skills in learners who do not have access to a resident teacher.

Diagnosis subroutines might enhance the effectiveness of drill and practice programs in several ways. First, the feedback that is provided can be made specific to the error that the learner seems to be displaying. Second, identifying the error the learner is making may prevent that error from becoming entrenched in the learner's cognition. Third, specialized feedback may give the learner more control over his/her own learning. Currently available drill and practice programs do provide feedback to the user on whether a particular response is correct or incorrect, but if the learner can't identify the reasons for the incorrect responses, there is typically little help provided for this purpose. In such a situation students tend to practice whatever skill they know rather than practice precisely the skill intended for use by the designer of the program.

A diagnosis program is created in several steps. (For more details on this process, see Bright, to appear.)

First, a clear delimitation of the content to be diagnosed is important, so that mistakes to be expected from students can be identified. Second, curriculum practices for teaching the content must be identified. Third, expected mistakes must be identified. Fourth, diagnoses must be created from the analyses of the wrong answers. Fifth, remediation suggestions must be made to correct the error(s) that the student seems to be making. Sixth, records must be kept of the responses that the student makes so that the teacher can follow-up on the diagnosis and remediation. The remainder of this paper will focus on the fourth step in this process.

Let us assume at the outset that the drill and practice program deals with a skill for which a series of exercises are randomly generated; for example, multiplication of decimals. Further, let us assume that a list of possible errors has been identified that students might make in responding to a particular exercise. For example, in solving 0.03 × 0.002, the student might type 0.0006. This mistake might reflect the error "counted all zeros to the right of the decimal point in each of the two factors instead of counting all the decimal places to the right of the decimal point."

It is essential that each error be computable, in the sense that if that error is applied to a particular exercise, the projected answer can be computed by the program. Computability is needed so that the response of the student can be compared to the predicted mistakes for the particular errors being diagnosed.

The overall thrust of diagnosis subroutines is to compare the student's responses to the predicted mistakes and then to look for patterns in these comparisons. For example, if a student's responses match error X in nine of ten exercises, and match no other error even four times, it is reasonable to infer that the student is using the incorrect process associated with error X. That is, if error X predicts the responses of the student and no other error seems to do so, it is reasonable to believe that error X is the difficulty the student is having. The student might be told what his/her error seems to be or might be given a substantive hint designed to point out the flaw in the process associated with error X. The two primary subroutines, then, are (a) generate the comparisons of the student's responses to the predicted mistakes and (b) create the diagnosis out of all of those data. The first of these subroutines is straightforward once a decision has been made regarding the way that the data will be stored. There are two obvious options. The first option is to maintain a complete matrix of all matches (1) and non-matches (0). Each row in the matrix represents the comparison of the student's response to a particular exercise with all the predicted mistakes for that exercise. Each column in the matrix represents the comparison of all the student's responses with the predicted mistakes for a single error in the list.

The second option is to maintain a tally for each error of the number of times the student's responses matched the predicted mistakes for that error. This option obviously does not maintain as complete information as the first option, but it may maintain enough information for practical purposes.

Typically not every error can be applied to every exercise, particularly if the exercises are randomly generated. Some errors require that specified conditions be met by the exercise before the error can be applied. For example, students might confuse the algorithms for multiplication and addition of decimals; when the number of decimal places is the same in both factors, students might use that common number as the number of decimal places in the product. In the exercise, 2.3 × 4.2, students might write 96.6 instead of 9.66 as the product. This error would not apply to the problem, 5.4 × 6.33, however, so whether the error applies must be stored by the program.

Possible pseudocode for this subroutine is given below. The number of errors to be checked is NUM; SR denotes the student's response; PA(I) denotes the predicted answer for error number I.

```
If SR is incorrect
  Set ctr = 1
  Do while ctr <= NUM
    Generate PA(ctr)
    If SR = PA(ctr)
      Set matrix entry = 1
    Else
      Set matrix entry = 0
    Endif
    Set ctr = ctr + 1
  Enddo
Endif
```

In the course of presenting exercises to the learner, the matrix must be searched periodically (for example, after every third exercise) in order to know whether the student is responding in a way that would suggest that a particular error is being made consistently.

One of the difficulties in conducting this search is that some responses may not match any of the errors identified beforehand. Rows in the matrix associated with these responses will contain only zeros. It may be advisable to delete such rows, if there are not too many of them, at least for the purposes of providing a diagnosis for the learner. Of course in the context of development of the subroutines, these rows provide important information; namely, that the creator of the subroutine has not analyzed the potential errors adequately. Personal interviewing of the student may be necessary to identify the particular unexpected error that is being made.

Pseudocode for the diagnosis generating subroutine is given below. The first DO loop deletes rows of the matrix that contain only zeros. The last DO loop generates the feedback to the user on the error that is being committed. RCTR and CCTR indicate the row and column, respectively, of a particular matrix entry; NEX denotes the number of exercises; NUM denotes the number of errors to compare to each student response.

```
Set RCTR = 1
Set flag = 0
Do while RCTR <= NEX
  Set CCTR = 1
  Do while CCTR <= NUM
    If entry (RCTR,CCTR) <> 0
      Set CCTR = NUM + 1
      Set flag = 1
    Else
      Set CCTR = CCTR + 1
    Endif
  Enddo
  If flag = 0
    Delete row RCTR
  Else
    Set flag = 0
    Set RCTR = RCTR + 1
  Endif
Enddo
Set CCTR = 1
Do while CCTR <= NUM
  Sum entries in col(CCTR)
  Compute (column sum)/(number
    of opportunities for
    committing the error)
  If ratio > cut off value
    Give feedback on error CCTR
  Endif
  Set CCTR = CCTR + 1
Enddo
```

Once it has been determined that a particular error has been made, remediation needs to be given. The help should be direct and to the point but must not divert attention too greatly from the main purpose of the program. Much more needs to be known about remediation before this can be provided with universal effectiveness. Until that time, programs may be limited merely to telling the learner that a particular error has been committed.

Computer users should begin now, however, to demand that diagnosis become part of drill and practice. Hopping (1983) has provided some guidelines for examining software that diagnoses students' difficulties outside a drill and practice setting, and she has identified some of the currently available software to accomplish this goal. Careful study of that software might suggest alternative ways of organizing the diagnosis subroutines.

REFERENCES

Bright, G. W. Computer diagnosis of errors. School Science and Mathematics, to appear.

Evans, C. The Micro Millennium. New York: Washington Square Press, 1979.

Fisher, C. W., Berliner, D. C., Filby, N. N., Marliave, R., Cahen, L. S., & Dishaw, M. M. Teaching behaviors, academic learning time, and student achievement: An overview. In C. Denham & A. Lieberman (Eds.), Time to Learn. Washington, DC: National Institute of Education, 1980.

Hopping, L. Guidelines for finding and using diagnostic/perscriptive software. Electronic Learning, 1983, 2(4), 76-78.

Okey, J. R. Diagnostic testing pays off. Science Teacher, 1976, 43, 27.

Travis, B. P. The diagnosis and remediation of learning difficulties of community college developmental mathematics students using computer technology (Doctoral dissertation, The University of Texas at Austin, 1978). Dissertation Abstracts International, 1978, 39, 2115A. (University Microfilms No. 7817722).

Woerner, K. L. W. Computer based diagnosis and remediation of computational errors with fractions (Doctoral dissertation, The University of Texas at Austin, 1980). Dissertation Abstracts International, 1980, 41, 1455A. (University Microfilms No. 80212529).

THE FLORIDA INFORMATION RESOURCE NETWORK (FIRN)

David L. Brittain
Administrator, Educational Technology
Florida Department of Education
Knott Building
Tallahassee, Florida 32301

ABSTRACT

The State of Florida has developed a plan for the implementation of a comprehensive data communications network that would link all public educational entities. A pilot project that is designed to explore the feasibility and benefits of such a network is currently underway. Contingent upon the success of this pilot, full scale implementation of the network is scheduled to begin in January, 1984.

FIRN

In response to a Legislative mandate, a plan has been prepared for the implementation of a data communications network that would support Florida's public education system. The proposed network is called the Florida Information Resource Network (FIRN). If fully implemented, the network will include data communications links connecting every public educational facility in the state.

An educational computing network has been the topic of many discussions since the mid 1970's in the Florida Legislature and Department of Education. However, with one major exception, these discussions were not followed with any substantive action. The only networking undertaken occurred within our State University System. In an effort to conserve funding of computing hardware, software and technical personnel, all nine state universities were linked to four regional computing centers as a means of providing administrative computing support to the entire university community.

In 1977 the Legislature began its initial networking efforts when it provided funds to promote an adequate computing capability for every public educational entity. From 1977 through 1981 approximately $5,000,000 was appropriated to promote resource sharing among educational units within the system. Essentially, these funds were used to assist with the acquisition of computing software and hardware that would be shared among two or more educational users.

A typical sharing arrangement supported by these funds was the connection of one or more small educational users (e.g., school districts) to a remote computing facility operated by another, larger educational entity. These efforts began to convince educational administrators that a remote computer could be a satisfactory substitute for one's own system. Resistance to the concept of networking began to disappear.

With this as a backdrop, the Spring of 1981 found the State University System searching diligently for ways to enhance its network which, over 6 to 8 years had evolved into a maze of patches on top of patches. At the same time, school districts were seeking ways to effect the automated submission of state reporting information to the Department of Education. It soon became clear that the solution to the problems of both groups was the same: a state-of-the-art data communications network serving school districts and state universities.

To address this requirement, the 1981 Legislature assigned to the School District Council on Comprehensive Management Information Systems (SDCCMIS) the task of developing a plan for a Florida Information Resource Network. Staff of the Department of Education and the State University System were to provide assistance to this group. Following many hundreds of hours of personnel time devoted by school district representatives and state agency staff, a plan was submitted to the Legislature in January, 1982.

Often school district and university staff do not work well together, however, a very interesting thing happened to these two groups as they worked on this project. The suspicions which each group had for the other began to disappear; to be replaced by a mutual respect for the expertise of the other. The effect was a resourceful and productive planning team which developed an outstanding plan and used less than 60% of the funds appropriated for that purpose.

The assignment of the responsibility for developing the FIRN plan to educational users was based upon our experience which indicated that the "sledge hammer" approach would not work. In other words, resistance to the network by school districts and colleges would be very great if it was forced upon them. Furthermore, since educational users are the ones who must live with the network, they are in the best position to identify the approaches which will insure success. Their involvement also creates a pride of authorship -- a sense of owning the network and a determination to see it successfully implemented.

Although educational users are involved in network planning and implementation, it is still necessary to provide inducements to obtain full participation in the network. One inducement is to emphasize the benefits which will accrue to educational users that become partners in the network. Another inducement is a mandate, levied upon school districts, which requires that certain data be reported in an automated form. Efforts by districts to develop the most effective manner of complying with this requirement lead to the realization that the most effective way to comply would be through a network. Another inducement, admittedly more forceful than the others, is the requirement that the Commissioner of Education monitor progress of districts as they develop means of submitting data in an automated form. For districts not making satisfactory progress toward achieving this objective, the Commissioner is to inform school boards that they are to withhold the salary of the superintendent until such time as adequate progress toward meeting this objective is being made.

The FIRN plan calls for a phased implementation of the network. First to be provided will be a batch communication capability for transmittal of summary data between school districts and the state. It is planned that this will be operational by June 30, 1985. Interactive access to summary information at the school district level is scheduled for implementation by June 30, 1987. Interactive access to summary information at the school level is scheduled for completion by June 30, 1989.

Before appropriating the first portion of the $46,000,000 which the plan says is needed for full implementation of FIRN, the Legislature felt a pilot project which was proposed in the plan should be conducted first. Accordingly, $529,000 was appropriated to enable the Department of Education to explore the feasibility of such a network.

The pilot project, which will be a series of thorough tests of the network, will be conducted throughout 1983. During this period at least 15 school districts, all 9 state universities, 3 community colleges and the Department of Education will have access to the network. These units will participate in varying degrees in the pilot. Among the more interesting experiments to be conducted are the following:

1. The automated transfer of FTE (full-time equivalant) student data from a school district to the Department of Education;

2. The access of student record software located in a large, urban school district (Miami), by a small, rural high school (Quincy) 500 miles away;

3. The transfer of transcripts of all high school graduates of a large school district (Dade) to a large community college (Miami-Dade).

Assuming the pilot is successful (and the technical staff is very optimistic that it will be), additional funding will be available so that full implementation of the network can begin. At this point it must be emphasized that merely installing the hardware will accomplish only a few of the planned benefits. To satisfy most of the aims of FIRN, significant applications software efforts must be expended. The plan identifies this as a significant problem area and specifies resources needed to address it.

Another area of concern to all those involved in the project is the need to insure the privacy of individual records. Every known means of protecting against security breaches will be employed. Some of these include limitations on the number of terminals that can access various data bases; password protection for all data bases; and prohibitions against accessing identifiable data except by authorized staff of the district, college or university that owns the data.

Superimposed over the technical and administrative complexities of the project are significant political considerations. Creating a facility that will provide benefits to the state, districts, colleges and universities without infringing upon the "local autonomy" of these entities is a most difficult assignment. When the concerns of funding, governance and "territorial rights" are added to the autonomy issue, it is obvious that the most difficult problems to be overcome are political in nature.

Those who are involved in the development of FIRN are convinced that an excellent plan has been prepared and that progress is moving ahead on schedule. Helping to confirm this belief in the plan is the fact that 12 to 15 other states, 2 or 3 Canadian provinces and at least 2 other nations have requested copies.

One comment regarding the impact of the desk-top or microcomputer on FIRN should be made. Like everywhere else, Florida is experiencing a tremendous growth in the use of these systems for educational purposes. As a matter of fact, Florida is recognized as a leader among the states in the use of these types of computers in its educational programs. It is probable that many, if not all, of these small computer systems will eventually become entry points into FIRN. In that mode they can be used to support instructional programs in the schools; gain access to instructional software that will reside in large storage units in regional centers; and, provide a means to communicate directly with other educators anywhere in the state.

In conclusion, the following are some of the benefits of the information resource network:

1. Automated reporting will result in more accurate and timely information which, in turn, will facilitate more effective decision-making for legislative and administrative purposes.

2. Access to some Department of Education data bases will be provided to educational users throughout the state. Some of these include teacher certification files, the higher education course numbering data base, and student financial aid.

3. FIRN will provide for the delivery of educational services such as instructional support provided by PLATO and career guidance counseling such as that supplied by the CHOICES and DISCOVER Systems.

4. The network will extend access to computing resources. Educational computing resources from any regional computing center will be available for use by any educational user, regardless of size or location.

5. Communications costs will be better controlled because communication lines will be more effectively used and state-of-the-art communications media will be employed.

6. Automated recordkeeping will substantially reduce the data burden on both administrative and instructional staff. For example, information on students will be captured at the time of their initial entry into the educational system and forwarded to each successive educational institution which the student enters. The same information won't be entered year after year as it is now.

7. The data transmission capabilities will greatly facilitate the routing of student records between elementary and secondary schools, between high schools, community colleges and universities, and between districts when K-12 students move.

8. Individual communications will be enhanced through the availability of an inexpensive electronic mail facility. Similarly, the availability of an educational electronic bulletin board will further enhance communications among educators.

9. FIRN will reduce the problems associated with non-compatible computer systems and terminals being used by various educational entities.

10. This facility will provide a model which may be employed by other state agencies to address data communications problems. Very substantial savings could be realized if one network such as FIRN could support most or all state agency communications needs.

The plan specifies a cost of $46,000,000 for full implementation. It will have to be implemented for less than that. However, with the growth and economic expansion projected for Florida, the Legislature must allocate resources sufficient to accomodate this growth. The educational system must receive priority attention. Within that context, everything possible must be done to insure the availability of adequate technological resources to accomplish Florida's educational goals. The Florida Information Resource Network is one such resource.

PLANNING FOR COMPUTER ACQUISITION

Raymond F. Brogan
Assistant Professor, Data Processing
Piedmont Virginia Community College
Rt. 6, Box 1-A
Charlottesville, VA 22901

ABSTRACT

The first years of the computer era were hectic. Mistakes were made. Millions of dollars were spent on inadequate equipment and unneeded systems. False starts were common because media-hype, pressure from sales people or a "keep up with the Joneses" attitude governed acquisition decisions rather than logical, rational planning. Today, the micro computer era is in the midst of the same hectic environment. We must learn from the past and avoid repeating the errors of the first decade by planning for our computer acquisitions.

The computer has been around since 1951 when the first commercial computer was installed in the Census Bureau at Suiteland, Maryland. It was called UNIVAC I, took up a considerable amount of space, was hot, cumbersome, and by today's standards it was rather erratic in its computations. It cost several million dollars.

Thirty years later I purchased my own computer. It sits on my desk top, is cooler than my TV, is neat and compact, has several times the computing power of the UNIVAC I, and cost three thousand five hundred dollars.

Between 1951 and 1981 there were billions spent on computers by government agencies, large corporations, educational institutions, airlines, motel chains, and many other groups. In the last three years there were over $4.9 billion dollars spent on personal and micro computers by the general population, small businesses and school systems. If you believe the commercials on TV and radio, the dawnings of the age of Computerius is upon us. The majority of people believe that it will raise our living standards and the educational level of our children. There are over 100,000 computers in our schools, approximately one computer for each 400 pupils. Several universities, Carnegie-Mellon, Drexel University, and Virginia Polytechnic Institute among others now require or will require entering freshmen to have their own personal computer.

What has caused this fervor for the computer?

The first commercial computer was manufactured by Remington Rand when it was an office furniture and equipment manufacturer. Fifty units were manufactured and when Remington Rand was urged to advertise, the then president of the company is reported to have said, "We have the best computer on the market. Anybody who needs a computer knows it. If they don't know it, they don't need a computer." He was probably right. On the other hand, IBM's approach was - we make computers and to stay in business we have to sell computers, so they advertised and hired sales people by the platoon and sold computers by the hundreds.

The first decade of the computer was hectic. Newer and better models were produced every couple of years. It took two years from day of order to day of delivery. Your new computer was obsolete before you got it installed and the waiting time between order date and start of manufacture date grew longer and longer and in some cases it was as long as three years from order date to start of manufacture. Add on the actual construction time and you had to wait five years for delivery date.

What caused such a demand? It was caused by a mix of sales push by manufacturers, media-hype by so-called experts, and a get on the bank-wagon attitude on the part of purchasers. Millions of dollars were spent on computers that never achieved their full potential because of a combination of ignorance on the part of purchasers who bought equipment they didn't need, and agitation from a new breed of business professionals - The Data Processor. These were heady times. By 1961 we were into the second generation of computers. The old, cumbersome, hot, erratic electron tube machines with their tons of air conditioning and miles of cabeling had given way to slick, cool, accurate, transistorized machines with considerably reduced requirements for air conditioning. In 1964 the third generation was born with integrated circuitry, multiprogramming, remote input and output. It was almost 1970 before supply began to meet the demand.

By this time there were also other manufacturers catering to specific needs of computer users who couldn't afford the expense of buying their own computer or who had needs that were not general enough to make it profitable for the big companies to handle. The mini computer was a product of this development and it began to reduce the cost of data processing from millions of dollars to hundreds of thousands of dollars.

By the beginning of the third decade of the computer the frantic growth of computer development had slowed down. There was one giant in the field recognized and envied by the other eight or ten major manufacturers who shared the twenty-five percent of the market that they had managed to obtain in the often bitter struggle for existence.

During the first two decades there developed a new breed of business specialist - the Data Processor. In the late fifties, if you knew what a computer was and were articulate enough, and wrote well and often, you could be - in fact you were - a computer expert. Top and middle management were at the mercy of the System Analyst who dictated the data processing needs of the establishment. Many a corporate executive, Navy Admiral, Army General, were told what to do by this new junior executive and did it. Millions of potential profits became debits because of ill-advised purchases of equipment and software systems. Today, in the fourth decade of the computer era, data processing is at ease with its environment. Top management now has its share of the brash pushers of twenty-five years ago, mellowed and scarred by the Hardware, Software, Information Systems wars.

But there is a new wind blowing. "A source of new life has been delivered to us," to quote Time magazine in its Computer of the Year issue. Parents and teachers are cowering in awe of the ten-year-old super genius who can program a classroom computer. Traditional education is being forced down new paths. "A computer can teach reading four times faster and 40% cheaper than a teacher," claims Control Data Corporation in a sales spiel for a software package called PLATO.

Is all this true or is it just another dose of advertising hype - a reoccurrence of the end of the first generation? It probably is a bit of both. We are in a new age of computer development. Within the last year the cost of a true computer has dropped to less than $100. By the end of the 1980's they will probably be as much a part of a school child's equipment as the pocket calculator is today. But that doesn't mean that they will have contributed much toward education. In its day, television was touted as the new tool that would make the teacher obsolete and we know what a miserable failure that prediction turned out to be.

The new wave in computer manufacture is the result of the micro chip, the electronic speck that has brought the computer from the giant bureaucratic establishment to the dining room table. The equipment is ready and is getting cheaper by the week, (by the time you read this the price will probably have dropped several quantums from what it is as I write.) The hardware is available, cheap and fairly simple. What is missing and what will always be a problem is the software to use on the computer.

The first problem is simply trying to cope with the enormous volume of claims made by all the vendors of software. Anybody with a year or so of programming experience can write software packages that will do what the vendor claims they will do. But what good is a software package that only allows you to store progress reports for ten children on one disk when you have thirty pupils and should be able to store fifty reports on a single disk. Of what good is a software package that teaches first grade skills at a fifth grade reading level? Is a tutorial software package worthwhile if it takes fifty minutes to elicit three responses from a student because the tutorial is so wordy or the manipulating software so convoluted that the user is left wondering if it will ever get to the point?

The traditional problem in the computer field has been that software is generally written by people who are computer oriented rather than subject oriented. The problem is still with us. Anyone who can program can write and sell educational software. By the same token, a teacher who knows the educational side of the problem probably hasn't the programming skills to do a competent job.

The second problem with buying preprogrammed software packages is the same problem that defeated the Management Information Systems approach of fifteen years ago. The software is written to sell to the largest market, therefore, it is either so general that it is of little value unless costly customizing modifications are made, or everyone is forced to change their way of doing things to conform to the requirements of their software package.

How do we overcome these difficulties? Do we train all teachers to be programmers? And if we did, what programming language would we use. At present there is only one language, BASIC, that is in anyway universal and its usefulness in educational programming is questioned by many educators. There are other programming languages being developed. The classroom teachers cannot be expected to write their own software nor for that matter can they be expected to evaluate all educational packages that are available or that will become available. At the same time they cannot be totally ignorant of the computer and its applicability as a tool.

As far as the educational computer is concerned we are at the same crossroads that the business world was in the early sixties. The micro computer is now in its second generation; for less than twenty-five thousand dollars you can have a computer lab that will allow twenty students to use individual computers tied to a central unit.

The media hype is tremendous. Dozens of manufacturers are trying to sell their products to school systems, teachers and students. Radio and TV commercials for computer systems are as common as shampoo, soap and cosmetic commercials. Among educators, there is an increasing competition to become known as a computer expert. Programming techniques that were discarded twenty-five years ago by the Data Processing community are being discovered as "new" ways by the micro programmer. Computer systems are being sold with promises of tailored software which will never materialize. Computer systems are being bought solely to keep up with the Joneses. Computer systems are being bought by administrators who have little or no idea of how they will use the equipment.

It is not too late to profit from the mistakes made by the business world in the early sixties. If we are to make full use of the wonderful power of this electronic tool we must approach its acquisition logically and rationally rather than emotionally. We must develop a standard approach to follow in all hardware and software acquisitions. Do not be misled by size or price. A micro computer is a computer and its purchase should be approached with the same attitude as any other capital goods purchase.

Develop a Standard Acquisition Plan.

I. Be humble. Are you capable as an individual or as a group to evaluate your needs and match them against all the propaganda that is daily bombarding you? Can you truly state your needs and say that they can be solved by the ABC Computer or the Ezysue Software Package? If you cannot, then get help. If you do not have a truly qualified Computer Resources person available, create a Computer Resources position. Find a willing person and give that person time and opportunity to become qualified if they are not current with the state of the arts, and provide for continuing education. Do not invest acquisition authority in the resources person. The position should be advisory, but the person should be strong. Provide opportunities for Computer Literacy development for all personnel involved with the computer or software acquisition. IN THE MEANTIME, HIRE A CONSULTANT.

II. Define the needs that will be satisfied by the acquisition. Remember that both software and hardware are tools. Your choice of tools depends upon the job to be performed and the function of the tool. You should not use a chisel as a screw driver. The computer is not a self-contained tool. It needs other components to satisfy most users. A computer configuration for a classroom probably would not satisfy the needs of the school board.

III. Define the goals to be met. What specifically is to be gained by the acquisition? Administrative goals are not educational goals. Recording skill levels in reading is not the same thing as using the computer to help teach reading. What are the time constraints, who is involved, what are the intermediate objectives to be met in reaching the goals?

IV. Define specific expectations. Do you expect to save time or money? Do you expect the acquisition to fill in for the lack of people? Do you expect to teach reading four times faster at forty percent of the present cost?

V. Define the educational approaches you will take in Computer Aided Instruction. Will you use the Tutorials, Drill and Practice, Problem solving? If so, for whom and in what subjects, and on what levels? Will it be monitored self-study or directed group study?

VI. Define roles. Everybody involved or to be involved with the computer should know what is expected of them. The teachers should know their role in relation to the pupil and both in relation to the computer. The resource person whould know his role in relation to administrators and teachers. Above all, remember that successful computer utilization is a team effort.

REACTIONS TO THE ADMINISTRATIVE USE OF MICROCOMPUTERS IN SCHOOLS

Daniel J. Brown
Department of Administrative, Adult, and Higher Education
Faculty of Education
University of British Columbia
Vancouver, B. C.
Canada V6T 1Z5

This research addresses the question: what factors inhibit or facilitate the use of microcomputers in school administration? A three-stage change process is used as the conceptual framework. Interview data were gathered from a small sample of knowledgeable respondents in two school districts involved in implementing microcomputers. Results indicate a number of important inhibitors [such as illiteracy, computer phobia, overexpectations] and a set of facilitators [such as inservice, staff level decisionmaking and positive attitudes]. A model for change to automated processes is produced as a guide for decisionmakers.

One reason the study of the implementation of computers in the school office is important is that the number of potential applications is perhaps more extensive than any other area of education.

A most compelling reason for inquiry into the change process surrounding administrative computing in education is the need to formulate policy and procedures which will be effective at the district and school levels. In order for planned change to be effective, it must be based upon a knowledge of successful strategies that are relevant and specific to computer implementation. Even when no central office direction is given [as may be the case under retrenchment conditions], schools who make a commitment to the use of computers in educational administration still require suitable frameworks to guide their thinking about the change. See Brown and Reusse (1983) for issues of microcomputer cost-effectiveness. The present study was undertaken with the Berman et al. (1977) model as interpreted and augmented by Roberts (1978).

Local Process of Change Model

Robert's (1978:27) prime assumption is that in the process of innovation at the local level - for example the school level - a mutual adaption will occur. That is, both the innovation and the internal organization will change. She views change as a three stage process and identifies specific considerations in each stage which lead to the formulation of research questions with an explicit focus. a) In the mobilization stage, focus is placed on the stimulus for change, institutional and individual attitudes toward change, the generation of local support, and the decisions made on the nature and scope of implementation including the behaviors required of staff. b) During the implementation stage the mutual adaptation between the innovation and the internal organization takes place. The nature of these adaptations influence subsequent decisions and actions, and therefore data collection is particularly important during this phase. c) The final stage, institutionalization, occurs when the innovation is assimilated by individuals and incorporated into the system so that the change becomes no longer novel.

Roberts (1978:71) identifies four potential barriers to innovation which are particularly relevant to the application of microcomputers in school administration. First is the complexity of the computer itself. Second is resource requirements, where ample time and effort on the part of office staff are needed to put the microcomputer to work. Third is the demand for understanding. Computers are seemingly inscrutable to many principals and secretaries. And fourth is the need for role change - microcomputer use clearly implies sizeable alterations in the way in which school office personnel perform their tasks.

Methods

Two school districts initially working with administrative applications were included in this study. Both these districts were largely taking their own initiative in administrative use, but they each had recently received some assistance in the form of Ministry of Education allocation of Apple microcomputers as part of a limited, provincewide experiment.

It was decided to employ semi-structured interviews to gather the data on how the innovation was received. Borg and Gall (1980:313) assert that the semi-structured form "provides a desirable combination of objectivity and depth . . ." An attempt was made to elicit the views of all those in, or potentially in, key roles with respect to this innovation. The twenty-three subjects interviewed included secretaries, classroom teachers, counsellors, computer resource personnel, school building administrators, district supervisory staff, and district administrative staff.

A Precis of the Results

Personal Attitude: Eighty-seven percent of the respondents indicated positive personal attitudes toward the innovation. It was assumed that different experiences and levels of expertise might produce varying attitudes toward acceptance of microcomputers. However, the only differences among respondents due to these factors was that the greater the expertise, the more thorough and cautious were answers regarding the scope and pace of change.

Advantages: Eighty-seven percent of the respondents identified two factors as the primary advantages to the application of microcomputers for administrative tasks. The first was time saved due to the speed of data processing.

The other major benefit identified was in the versatility and flexibility of the programs themselves. Parents have displayed a very positive attitude toward the prompt and accurate information sent home about attendance, and some students have indicated a tendency to accept the veracity of computer print-outs over manually prepared documents.

Disadvantages: Sixty-five percent of those interviewed cited negative attitudes as the major disadvantage. Although very few of the twenty-three interviewees manifested negative attitudes themselves, they reported that others held these attitudes. The most frequently mentioned example was the frustration and impatience displayed when "the computer" did not function as anticipated. Knowledgeable respondents stressed that those who are not computer literate have no conception of what a microcomputer can, and more importantly, cannot do! The second most frequently mentioned disadvantage (43%) was lack of immediate accessibility to stored data due to demands on computer time from other programs or maintenance difficulties. Security of information was also seen as a problem by some respondents while others felt that it is no more of a concern than it is when using traditional methods.

Opportunities and Incentives: Sixty-five percent of the respondents from both districts identified the district helping teacher, or coordinator, as an important factor in providing opportunities to learn about microcomputer operations. These resource personnel are also seen as essential to successful implementation of programs in individual schools. They are needed not only to assist with programing and operator training, but also to be on call to debug new programs. Other opportunities identified included availability of inservice in out-of-school time, day workshops, and university courses. Also mentioned frequently was the personal incentive to be at the leading edge of such a pervasive new technology.

Resistance: Computer illiteracy and computerphobia were seen as the two principal reasons for resistance to the innovation on the part of school personnel. This concern is also addressed in Harper and Stewart (1983).

Major inhibitors of change:

1. Computer illiteracy: lack of knowledge of the functions that can not be performed, lack of appreciation of the required time and skill to program functions that can be performed.

2. Computerphobia: fear that computers will dehumanize the teacher/student relationship, fear of role change and job stability.

3. Initial cost: the perception that initial hardware and software costs outweigh potential benefits.

4. Technical Problems and Security: equipment does not always function adequately, use of the new technology raises questions about how the information is to be protected.

5. Shortage of trained personnel: personnel with programming skills and educational experience are in short supply, as is the time required for such personnel to develop programs for particular school needs.

Major facilitators of change:

1. Speed and accuracy: microcomputers can process large amounts of routine data rapidly and accurately.

2. Versatility and flexibility of some programs: Those programs which are compatible with school needs permit considerable flexibility and are adaptable to different schools' organizational structures.

3. Staff-level decision-making supported by administrators: acceptance of change is greatly enhanced when the whole staff has input into the decision making process.

4. In-service: formal in-service programs and informal training from an on-site helping teacher increases computer literacy.

5. Attitudes: attitudes are improving due to the success of instructional applications and the pervasiveness of the technology in society.

Policy -- An Ideal Model

Initiation

1. Stimulus: Many factors from within and outside of the school system create a felt need for change. These factors include difficulty in coping with complex tasks, cost incentives, societal changes, and awareness of technological innovation.

2. Decision-making: School personnel begin mobilizing efforts for change when decisions are made at the staff level. Imposed decisions are relatively ineffective in initiating change.

3. Communication: Both advantages and limitations to new technology are introduced at this stage. Communication is facilitated by committee structures with representatives throughout the schools, and by a full time resource person.

4. Support: Support from staff, site administrators, and key district personnel is generated at this stage. Perceptions of lack of support inhibit innovation.

5. Preparation: Plans to implement are finalized. Clearly delineated goals, generous time projections and feedback mechanisms to monitor change facilitate subsequent implementation. Programs compatible with school needs are selected.

Implementation

1. Assistance: Operators are trained by resource personnel. On-call, and preferably on-site, expertise is required during implementation start-up.

2. *Limitations*: Initial feedback limits the scope and pace of change. Gradual implementation of one program at one school facilitates change.

3. *Adaptation*: As technological difficulties are resolved the innovation gains wider acceptance. Communication with parents facilitates acceptance.

4. *Expansion*: Attitudes develop based on relative degree of success. Positive attitudes lead to requests for new programs at the school site, and for participation in the innovation at other schools. Negative attitudes arise when goals are not met within projected time frames.

Institutionalization

1. *Assimilation*: When technological difficulties are resolved, individuals assimilate change and the innovation is incorporated into the system.

2. *Stabilization*: Mature programs are in place. The need for extensive backup systems is reduced.

References

Berman, P. et al. *Federal programs supporting educational change*. Vol. 7, Santa Monica, Calif.: Rand Corporation, 1977.

Borg, W. R. & Gall, M. D. *Educational research: an introduction*. 3rd edn. New York: Longman, Inc., 1979.

Brown, D. J. & Reusse, M. User perceptions on the cost efficiency of microcomputers for school administration. Mimeo, Faculty of Education, University of British Columbia, 1983.

Harper, D. O. & Stewart, J. H. (eds.) *Run: computer education*. Monterey, Calif: Brooks/Cole, 1983.

Roberts, J. M. E. *Implementation of innovations in educational organization and instruction*. Research for Better Schools Inc., 1978.

A TELIDON - APPLE II INSTRUCTIONAL MODULE

Robert Bruce, Steven White and Seong-Soo Lee
Education Research Service Centre
University of British Columbia
2121 Main Mall
Vancouver, British Columbia, Canada V6T 1Z5

ABSTRACT

Telidon provides an efficient, inexpensive colour graphics display protocol. By creating pages of graphics (and text) using the Telidon format, and modifying a page display program, an Instructional Module on Boyle's and Charles' Law has been developed. This module provides a tutorial followed by drill and practice exercises on Boyle's Law and another tutorial with drill and practice on Charles' Law. The tutorials incorporate Telidon pages which are displayed based on the user's responses. The module thus represents an instructional unit which synthesizes the emerging Telidon technology for graphics with the Apple II microcomputer.

INTRODUCTION

Telidon is the protocol for defining picture elements which was developed by the Canadian Department of Communications in order to meet the requirements for an inexpensive, compact method of describing graphics images (1,2). Using one of a number of picture creation systems, Telidon pages (pictures, or frames) are developed and stored for later display on terminals capable of decoding the Telidon protocol for the PDIs (picture descriptor instructions) which make up the page. At ERSC (Education Research Service Centre), we have created an instructional module about Boyle's and Charles' Laws, which uses Telidon pages for the graphics component of the instruction. Our motivation was twofold: first, to bring together the hardware and software required to support the development of computer assisted instruction which incorporates Telidon graphics, and second, to create a sample instructional module as an example for faculty and students in the Faculty of Education. Boyle's Law and Charles' Law were chosen as subject material for the module in order to present, graphically, topics which are usually treated in a mathematical manner.

HARDWARE AND SOFTWARE

A Microtel VTX202 Telidon terminal operating at 1200 baud was used to display the Telidon pages, both during creation and editing, and for presentation of the module. Pages were created using the VTX202 connected to several different computers; an Apple II+ using page creation software from ITG Inc., an IBM Personal Computer using page creation software developed by the Computing Centre at UBC (University of British Columbia), and an Amdahl V8 using page creation software developed by the Library Systems Group at UBC. The instructional unit, modified and expanded from the page display software by ITG Inc., is written in Apple Pascal and runs on an Apple II microcomputer with 64K of RAM.

INSTRUCTIONAL MODULE

The Instructional Module consists of a menu with four sub-sections, two on Boyle's Law and two on Charles' Law. One sub-section for each gas law consists of a number of Telidon frames which present using graphics and text, a tutorial about the gas law. The user can, at his own speed, move forward or backwards through the sequence of frames one at a time and can at any time exit to the menu of sub-sections. After viewing the Telidon frames, the user performs an experiment, using the Apple II, in which he is prompted for the value of either a gas volume (for Boyle's Law) or a gas temperature (for Charles' Law) and is then given the computer determined gas pressure. Invalid responses may terminate the experiment (as in real life) forcing the user to repeat the measurements. Upon completion of the experiment, the gas law equation is determined and presented graphically, and the user is returned to the sub-section menu. The other sub-section, for each gas law, provides a drill and practice session by presenting a user-selected number of problems using the gas law, and summarizes the user's performance as a percent correct at the end of the drill.

RESULTS

Our experience using several different page creation systems led to several conclusions regarding page creation in general, and the individual systems which we used. Since Telidon terminals are currently low resolution devices, pages which contain stylised, simplified images are more easily produced and viewed than pages containing detailed diagrams. The ability to group Telidon primitives (fundamental picture elements such as: point, line, circle, arc and polygon) into an item which can be referred to by name, edited, moved, etc.,becomes more important the more pages one creates. the ITG software could only edit Telidon primitives, and thus was tedious to use for pages which contained more than a few elements. One of the most striking aspects of developing pages is the ease and speed with which major changes can be made to a picture. A good example is the marked effect on the appearance of the page made just by changing the background colour. In fact, with Telidon, use of colour becomes a significant aspect of the presentation. Some studies have appeared on the use of colour in CAL (3) and Telidon provides a system for further research in this area. Another factor which influences the visual effect of Telidon pages is the speed of production of the frames (i.e., the baud rate of the Telidon terminal). Using an Apple Communications Card at 300 baud, pages appear at a frustratingly slow rate. At 1200 baud the rate of display becomes a distraction only for complex or detailed frames. Depending on the Telidon terminal and the communications interface, pages can be displayed at up to 9600 baud at which point the speed of the system no longer interferes with the presentation. In practice, the dynamics of Telidon page display can be incorporated into the design of the pages. While the final product is a static picture, the noticeable time dimension of frames displayed at 1200 baud can be used to control the order in which the various components of the page are displayed, giving the pages an animated quality. Although the Telidon frames used in the Boyle's and Charles' Laws modules are static once they have been displayed, pages which are "real time" and changed based on user input have been created on mini-computers such as DEC VAX (4). In future work with Telidon we plan to investigate the possibility of creating frames which can be modified interactively using micro-computers.

The standard DOC Telidon page access system, a hierarchical tree structure used for random information searches, is too uncontrolled to be used in an instructional module. The ITG page display system permits the designer to specify, for each page, which other pages it can be accessed from and has access to, allowing control over the order in which the user proceeds through the instruction. This system works very well for displaying Telidon pages (the tutorial part of the module) and forms the basis of the instructional unit. By creating sections of code which use the Apple II instead of the Telidon terminal, the page display system can move the user from Telidon pages to programmed instruction and back. Since Pascal promotes programming in modular sections, it was more efficient and easier to write the units for this application in Pascal than in, for example, Basic or Fortran. More importantly, the modular design allows facile addition and replacement of both Telidon pages and programmed segments, encouraging improvement of this module, and the development of a "library" of frames which could be used in multiple applications.

CONCLUSIONS

As a result of this project, ERSC now provides our users with the capability of creating, editing and storing Telidon pages on microcomputers or mainframe computers, and transferring the pages between computers. The instructional module shows our users one use of Telidon in education, and hopefully, will encourage the development of other innovative applications of Telidon.

REFERENCES

1. Canadian Department of Communications, "A General Description of TELIDON: a Canadian Proposal for Videotex Systems", Technical Note No. 679-E, Ottawa, 1978.
2. Godfrey, David and Chang, Ernest. The Telidon Book. Press Porcepic, 1981.
3. Durrett, John and Trezona, Judy. "How to use color displays effectively,' Byte. April 1982, Vol. 7, No. 4. p. 50.
4. Cowper, W. Donald. "Telidon Documentation", Computing Services, Athabasca University, 1982.

LOGO K-6: THE DEVELOPMENT OF THOUGHT AND MEANING

MRS. DONA BURRON

RIVER EAST SCHOOL DIVISION #9

MR. EDWARD BULLER

DEPARTMENT OF EDUCATION, PROVINCE OF MANITOBA
PREVIOUSLY RIVER EAST SCHOOL DIVISION #9
589 Roch Street
Winnipeg, Manitoba
R2K 2P7

This session will describe and detail the necessary organizational and procedural models necessary for the successful implementation of Logo in the elementary classroom. The session will attend to both the prior academic conceptual understandings as well as the procedural details of implementation.

Prior understandings necessary include the place of Logo in the development of language, concepts in Mathematics, and problem solving skills. These will be dealt with by describing a series of teacher professional development workshops held during academic year 1982-83.

This will be followed by detailing the necessary practical parameters of implementation such as classroom management, materials and resources, teacher role, program planning, organization, and evaluation.

This session will utilize a variety of media and presentation techniques such as overhead transparencies, 35mm slides, and exhibits. Emphasis will be on the practical aspects of implementation; a booklet summarizing "Ways of Getting Started in Logo K-6" will be made available.

It is fascinating to observe the interaction between child and computer in a situation where the child is in control of the computer. An amazing thing happens - or is it so amazing? The child becomes so interested in the technology of the computer-that which is unique to it-that he puts tremendous thought and effort into figuring out what the computer can do and then applying this to suit his own purposes.

Because it is the nature of the child to attempt to understand and master his environment, Logo allows the child mastery of many cognitive skills and also allows him to manipulate powerful ideas through which he may gain some control over his environment. (This is what children do in their play-constantly manipulating and repeating ideas and concepts in order to gain mastery)

When Seymour Papert, together with his colleages at MIT, devised and developed LOGO, he presented a computer language which allows learning the language of communication in Logo in the same natural way that he learned the language he speaks. This language, like our spoken language, also contributes to the development of thought and meaning.

It is in this language learning environment that the child learns to use the computer in a masterful way. Papert, in his book Mindstorms:Children, Computers, and Powerful Ideas reminds us of the work of Jean Piaget in which he explains that children are builders of their own intellectual structures and need materials with which to build. Some of those thinking materials are unavailable to the young child because they are too complex and abstract.

It is not until later years that the child is abile to use them i.e. algebraic algorithms.

However computer technology and more specificly Logo, rather than attributing to the child the ability to skip certain developmental steps and thus handle very abstract material, has instead made the material itself simple and concrete. Now the child can handle it.

Because the Logo philosophy is that of self-direction and purpose it is differentiated from some of the other kinds of experiences children may have with computers where the computer directs the child for someone else's purpose. This is not to imply that other types of computer interaction cannot be valuable. It does imply however, the necessity of providing teachers with: a) an understanding of the philosophy of Logo that they might implement it properly.
b) enough fluency in the Logo language that they may fascilitate the students' development-i.e. by providing appropriate models. Thus teacher preparation is critical.

The professional development workshops on Logo with which we will deal consist of two parts. The first part is an introduction to Logo and the second part deals with curriculum design, implementation, and evaluation. Both are hands-on sessions.

In the later sessions some of the topics of concern include-goals and objectives of Logo, teacher role, uses of Turtle Geometry in the classroom, learning styles, planning and debugging, teaching mathematics through Logo, curriculum applications, and evaluation.

Because the learning activities in Logo are a departure from many traditional classroom approaches to learning, there are some concerns that need more specific attention. Justification of Logo activities in the classroom is one. To people unfamiliar with Logo, which may include anyone from the principal to parents to fellow teachers, the teacher using Logo has to be able to explain what children are learning and how it can be applied to the curricula (i.e. problem-solving/math) Also, the teacher needs to be able to evaluate children's work in Logo. This means showing teachers how Logo activities imply constant diagnosis and prescription. These then become evaluation tools, considering the goals of the Logo program.

The final session of the workshop is devoted to actual classroom implementation. A discussion concerning the ratio of students to computers is necessary because of the inconsistency in numbers of computers available to individual schools. Reference to the Logo philosophy narrows the ratio possiblities if the program is to allow for sufficient contact time.

Other concerns in initiating Logo in the classroom include - materials and resources, scheduling, record keeping, initial activities (pre-logo and orientation) group activities and discussion, fascilitating techniques and materials, and sharing and display of work.

Once a teacher has had preparation in these three areas - philosophy, introduction to Logo activities, and classroom implementation, Logo can be used in the classroom. The session will deal with all three areas but emphasis will be on practical detail. It is hoped that the session will not only provide a framework providing training for teachers in Logo but will also provide a starting point for getting started in the classroom. A booklet providing practical ideas and details will be available.

References:

Papert, Seymour. Mindstorms:Children, Computers, and Powerful Ideas New York: Basic Books, Inc. 1980

Final Report of the Brookline Logo Project.September, 1979

CHILDREN'S APPROACHES TO PROGRAMMING: A TEACHER'S PERSPECTIVE

George Burns
Bank Street College of Education
610 West 112th Street
New York, New York 10025

ABSTRACT
Children tend to accept programming techniques for their function without examining how the techniques fit into the logic of the language. This enables them to use certain programming techniques more easily than adults but ultimately prevents them from developing an understanding of the language.

In my work with eight and nine year olds at the Bank Street School, I try to introduce new concepts and ideas from a concrete perspective, whether the concepts have to do with math, science, social studies or language. All of the children in my class, even the brightest ones, benefit from taking hold of a thing in their hands and experiencing it before letting go and dealing with it in the abstract. Despite the fact that we are using Logo, a very graphic language, teaching these same children how to program the six microcomputers in our classroom has placed much of our work and discussion on a very abstract level with few concrete places for myself or them to hold onto. At first this seemed to be a problem only for me, since they do not distinguish between concrete and abstract - they either understand something or they don't. The abstract nature of their computer work was not inhibiting their desire to program. It eventually became clear, however, that their approach to the abstract nature of programming was inhibiting their ability to really understand what they were doing on the computer. It also became clear that my methods of teaching programming were encouraging confusion rather than clarity in the children's programming concepts. To describe this development more fully it is necessary to start at the beginning of the Bank Street Logo Project.

When the computers were first introduced to my class, I wasn't surprised at how excited the children were, but I was amazed at how quickly they learned the Logo commands and used them in their work. They learned in a week what I had planned to teach them in month. One of the first things I noticed about their work was how they centered their attention on the computer itself. Information sheets designed to help them with programming were held onto only as long as it took to read them and put the information into their computer work. They had no patience for the little manuals that I had made, and their impatience seemed very logical - the computer should know more about itself and be able to give the user helpful information.

Even though the children seemed to be learning things very quickly, I was hesitant to introduce new concepts. Teaching programming was very new to me and I had no way of knowing which concepts should be taught and how they should be introduced. I was under the impression from having read Papert's Mindstorms that Logo is a language that almost teaches itself, but I soon realized that I was still going to have to do a lot of teaching. The most concrete place to start seemed to be in their work itself, so I looked for children whose programs could benefit from new programming techniques.

I noticed an eight year old girl who was experimenting with the REPEAT command by using it with arbitrary number inputs (e.g. REPEAT 25 FD 68 RT 38) Every time she wanted to try new number inputs she would retype the whole command. I thought very cautiously that she might be able to make a program which would use variables for the inputs and explained this technique to her. Before I had even shown her how the program would run she asked if she could put as many variables as she wanted in her program. She then proceeded to put in a second REPEAT command with three more variables. Within the next two days she had experimented with this technique so much that it was necessary to introduce her to the concepts of recursion and conditionals. Her simple REPEAT statement had developed into this:

```
TO PG :A :B :C :D :E :F :COL1 :COL2
IF :COL1=6 MAKE "COL1 1
IF :COL2=6 MAKE "COL2 1
REPEAT 4 [PC :COL1  REPEAT :A [FD :B RT
  :C] PC :COL2  REPEAT :D [LT :E FD :F]]
PG :A :B+5 :C :D :E :F+5 :COL1+1 :COL2+1
END
```

Clearly she was able to play with more elements of the language than I had anticipated.

Another experience demonstrated to me further that the children were able to use more sophisticated Logo commands. Many children were writing programs which made sprites (animated objects on TI Logo) move in patterns across the screen. They seemed frustrated at having to edit their programs or type in new commands in order to change the pattern of movement. The READCHARACTER command (which enables you to make a program that responds to each key on the keyboard) seemed to be a perfect solution to their frustration. I hesitated to teach it to them, however, because of its abstract nature. To use it in a program you must set up a variable name under which is stored the information as to which key is being pressed. Then a set of conditional statements must be listed in order to run different procedures depending on which key is being pressed. Finally, the procedure must be recursive since the READCHARACTER command only stores one key. A short example of a READCHARACTER program looks like this:

```
TO CONTROL
MAKE "KEY READCHARACTER
IF :KEY = "E SETHEADING 0
IF :KEY = "D SETHEADING 90
IF :KEY = "X SETHEADING 180
IF :KEY = "S SETHEADING 270
CONTROL
END
```

Using this procedure would allow children to control the movement of their sprites by using the E, D, X, and S keys. I introduced this procedure to a small group of children who immediately implemented it in their programs. When other children saw that these children could control the movement of their sprites very easily, they asked to see the procedure and put it in their own programs.

In neither of these cases did the children express confusion as to how their programs were working. They did not ask questions the way most adults do about why you must use quotes in one place but a colon on another. What they knew was that a certain set of commands would make the computer do a certain thing, and in order for them to be able to make that thing happen in their programs they just had to remember the commands. They concentrated on what the commands did, what function they performed on the computer, and not at all on the command's place in the logic of the language.

By the end of the year it was clear that the children's functional approach to programming was preventing them from being able to come to a real understanding of the language. It was very difficult for most of the children to create complex programs on their own, even if they had all the pieces they needed. They always needed someone who knew more than they did to help them put the pieces together. Their concentration on function even obscured some of the concepts that could have been more easily available to them. For example, the girl who was able to fit variables into her program above, was never very clear about what each of the numbers was doing when she ran the program. Her program, though she was able to write it, was too dense to figure out, and she wasn't so concerned with that anyway. Instead she would run her program with arbitrary sets of numbers and write down the ones she liked. Her program had become more sophisticated but her conceptualization of what the computer was doing was no more advanced than when I first noticed her playing with REPEAT. In only a very few cases did children seem to have developed a sense of the general structure of the language. Logo wasn't teaching itself, and I wasn't doing a very good job either.

With Bank Street's support I spent most of last summer increasing my own programming skills and developing a more coherent method of teaching Logo concepts. This year rather than introduce concepts as children seemed to need them in programs, I have been teaching the whole group interesting programming techniques along with some fundamental principles of Logo's logical structure. Instead of explaining variables only in a utilitarian situation, we have spent several weeks looking at the differences between local and global variables, the uses of the command MAKE and the distinction between WORDs and LISTs. At each of these lessons I try to present the logical structure in a manner that is both appealing and understandable, and I have been pleased to find that the majority of the group is interested and takes part in these exercises. Some examples of these lessons should make this more clear.

As is evident in the case mentioned above, one of the most difficult things for children (and adults) to understand is what the value of the variable is actually doing in the program. I have used several methods of reinforcing this concept, one of which comes out of a very popular game for children. In this game a story has been printed out with several words missing. The words are only identified by the name of their part of speech. Without looking at the story aperson is asked to give words that will fit the particular parts of speech. The resulting story is usually ridiculous, and the lesson in parts of speech is learned easily and in an entertaining fashion. The computer can be programmed to play this game very easily, teaching not only the concept of parts of speech but also the concept of variables. After only a few lessons every child in the class was able to make a program of this kind:

```
TO STORY :NOUN1 :NOUN2 :NOUN3
PRINT (SENTENCE "A :NOUN1 [WAS WALKING
  DOWN THE] :NOUN2 [WHEN A] :NOUN3
  [FELL ON ITS NOSE] )
END
```

Running STORY "DIRT "HOUSE "GIRAFFE would make the computer print out:

```
A DIRT WAS WALKING DOWN THE HOUSE WHEN
A GIRAFFE FELL ON ITS NOSE.
```

In another lesson I might ask who in the class would like to be on the Yankees. On the computer I would then type MAKE "YANKEES [TOBY ABBY JOHN ANNA] I would then ask how they could make the computer print out John's name using the list YANKEES and the difficult functions FIRST, BUTFIRST, LAST and BUTLAST. It seemed clear they understood these words when they came up with this list of methods:

```
PRINT LAST BUTLAST :YANKEES
PRINT FIRST BUTFIRST BUTFIRST :YANKEES
PRINT BUTFIRST BUTFIRST BUTLAST :YAN-
 KEES
PRINT BUTLAST BUTFIRST BUTFIRST :YAN-
 KEES
```

In addition to giving more complete and coherent lessons in Logo consepts I have been more hesitant about taeching children techniques they don't ask for. This has meant that children's work has not advanced in great strides the way it did last year, but since they are not demanding to know more or expressing dissatisfaction with their work, it may also mean that children are very comfortable working at their own levels.

Children still focus, however, on what they want the computer to do rather than on why it works the way it does.

They also seem to separate the exercises they do in our lessons from their actual work on the computer in that they still need adult help in working through most programming problems. My hope is that when they are more developmentally ready in the years to come I will have prepared them with some conceptual background.

THE USE OF LOGO IN THE TEACHING OF GEOMETRY, K-12

Ralph W. Cain
The University of Texas at Austin
Austin, Texas 78712

and

Patricia A. Semmes
Trinity University
San Antonio, Texas 78284

ABSTRACT

The nature of LOGO makes it ideal for use in an interactive mode for the development of geometric concepts. The distance-direction nature of the motion of the turtle in turtle geometry makes it ideal for introducing and developing the concepts of closed curve, polygon, regular polygon, and vector. Ways to present advanced theorems and operations on vectors evolving from simple ideas that could be used by kindergartners are developed.

CLOSED CURVES AND POLYGONS

The computer applications presented here are based primarily on methods presented by Abelson and diSessa (1980) as modified for Apple LOGO. Geometric definitions are altered somewhat to relate them to the processes and images characteristic of the program used.

A simple closed curve may be thought of as a pathway in a plane that begins at a point, proceeds continuously, and returns to the starting point without having intersected itself at any other point. Turtle geometry is LOGO requires the turtle to move in straight segments, and, although arcs can be approximated by the use of very small increments of motion, the closed curves best suited for presentation in this mode are polygons. A polygon, in this approach, is a simple closed curve whose pathway consists of a series of straight segments. Since the connected segments form angles at their points of connection, the control of the direction of motion is necessary to the production of polygons. The ability of the user of a LOGO program to control both the direction and the distance of motion along a segment makes it possible to "draw" a wide variety of polygons by varying the length of the sides (segments) and the size of the angles formed, with the understanding that, eventually, the turtle must return to the starting point to complete the closed curve.

The simple commands available in LOGO turtle geometry are FORWARD, BACK, RIGHT, LEFT, PENUP, and PENDOWN. The last two control whether the motion leaves a trace on the screen of the CRT. To draw polygons the PENUP command is not useful. Also, FORWARD and RIGHT are sufficient to control the motion, since all BACK and LEFT commands are equivalent to some corresponding FORWARD and RIGHT commands. Therefore, we will limit the simple drawing of polygons to the use of PENDOWN, FORWARD, and RIGHT. Since polygons may contain sides of varying length, and the sides may form angles of varying measure, numerical inputs are used to determine the length of the segments and the size of the angle of direction change at the points of connection (vertices). The unit of distance is the "turtle step," and the unit of the change of direction is degree of angle. For example,

```
FORWARD 50 RIGHT 30 FORWARD 100
```

would trace a segment from some starting point, headed in an initial direction, a length of 50 steps, then turn right 30 degrees, then trace another segment of 100 steps. General polygons can be drawn by arbitrary selection of lengths and angles, with the conditions that the traced pathway cannot be crossed before the eventual return to the starting point. When one wishes to complete a polygon by returning to the starting point from the end of some side, it may not be so easy to determine by "eyeballing" how far to do or in which direction.

A technique that may be considered a first approach to the drawing of polygons is a program that allows for single keystroke control of distance or direction of motion. Such a program allows the operator to move forward in small increments and to change the direction of motion in small angular increments to the right (clockwise) by pressing a previously determined key for each. For example, the turtle may be made to move forward 10 steps by a single stroke of the F key, and the direction may be turned to the right 15 degrees by a single stroke of the R key. Thus, anyone from kindergartners to adults can draw a polygon on the screen with the limitations that sides must be of lengths in increments of 10 steps and exterior angles in increments of 15 degrees. Such an approach should aid in the final closing of a polygonal curve by allowing the operator to approach the starting point by closer and closer approximations. A feature that could be added is one which erases a segment in case one wishes to change its direction; that would help in the closing of the curve if it became apparent that the direction of the proposed final segment were clearly going to miss the starting point. Exact closure may not always be possible because of the limitations imposed by the increments, but very close approximations are always possible.

REGULAR POLYGONS

Of special interest are regular polygons, those in which all sides are of equal length and all angles are equal in measure. To draw such a polygon one could start with a FORWARD command with some predetermined input, followed by a RIGHT command with predetermined input, followed by the same FORWARD command, and so on until the polygon is completed (if one exists with the given inputs). However, the iterations feature of the LOGO system allows one to repeat a distance/direction change sequence indefinitely, a fixed number of times, a number of times dependent upon selected variables, or the sequence may be manually stopped. Since we are trying to construct polygons, the criterion for stopping the process should be when a polygon is completed or when it becomes apparent that one cannot be constructed with the given inputs. Consider the following program using the REPEAT command:

```
REPEAT 3 [FORWARD 50 RIGHT 120]
```

Such a program would draw an equilateral triangle with sides of length 50. (Note that the angle of the turn is the exterior angle of the triangle and is, therefore, the supplement of the interior angle.)

Another useful property of LOGO is that a procedure such as the one listed above can be given a name. For example, consider the following:

```
TO SQUARE

    REPEAT 4 [FORWARD 50 RIGHT 90]
```

We have given the name SQUARE the procedure that draws a square with sides of length 50. We can now use SQUARE to draw a square of side length 50 without any additional instructions. To draw squares of varying sizes, consider the following:

```
TO SQUARE :SIZE

     REPEAT 4 [FORWARD :SIZE RIGHT 90]
```

Then, a command of SQUARE 100 will result in a square of side length 100.

The procedure for attempting to produce regular polygons is called POLY. It draws a segment of predetermined length, turns right a predetermined angle, then repeats the sequence. At some point in this procedure the turtle will (1) return to the starting point, completing a regular polygon and will then, if not stopped, retrace the polygon, or (2) intersect the pathway, indicating that a regular polygon does not exist using the given inputs. The similarity of all regular polygons with a given number of sides suggests that the controlling variable for determining the existence of a regular polygon is the size of the angle and not the length of the side. The general program is:

```
TO POLY :SIDE :ANGLE

   FORWARD :SIDE

   RIGHT :ANGLE

      POLY :SIDE :ANGLE
```

Such a program allows choice of side length and angle of turn (exterior angle). Since regular polygons are convex and we have chosen to use RIGHT turns, it is wise to start the turtle in the left center of the screen, pointed directly upward. The length of side chosen is dependent upon the angle chosen; if the length is too great, the turtle may run off the screen (or go out-of-bounds) before the polygon can be completed or before the pathway is crossed.

We have already considered special cases in which ANGLE 120 and ANGLE 90 were used, resulting in an equilateral triangle and square, respectively. The program as shown would continue to retrace the polygon until stopped manually. What about other inputs for ANGLE? Geometrically, angle measures are limited to values between 0° and 180°, so we can limit our consideration to inputs for ANGLE in that range. Also, since very small inputs for ANGLE produce figures that look like circles, and since inputs close to 180 produce results that are difficult to interpret because of the resolution limts of the screen, we will limit our consideration to angles in the range 20 to 120.

If we try inputs for ANGLE of 20, 30, 40, 50, 60, 70, 80, 90, 100, and 120, we see that some produce regular polygons and some do not; specifically 50, 70, 80 and 100 do not, while the others do. What is special about those that do? To speed up a discovery--they are all divisors of 360, the number of degrees required to turn the turtle one complete revolution. Furthermore, the number of segments required to return to the starting point is 360/ANGLE. Thus, if the input to ANGLE is 40, a regular polygon of 9 sides is produced, and 360/40 = 9. Are there other inputs to ANGLE within our range of consideration that will produce regular polygons? How about 24, 36, 45, and 72? And outside our range are 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15 and 18. Also note that, if 360/ANGLE = SIDE, then 360/SIDE = ANGLE, so that regular polygons of any given number of sides can be constructed, at least to very close approximation, by dividing 360 by the desired number of sides, and then using the quotient as the input to ANGLE.

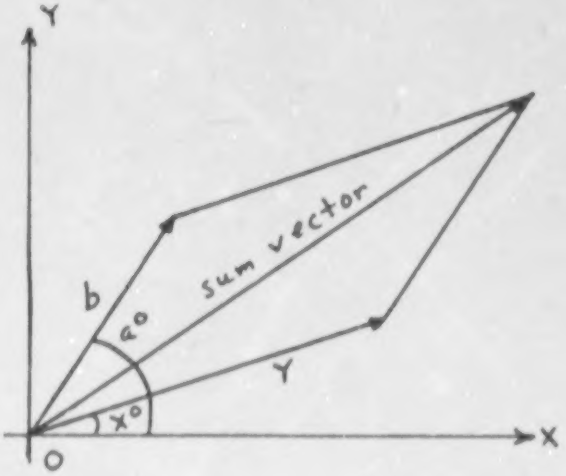
The preceding discussion is related to what Abelson and diSissa (1980) call the POLY Closing Theorem (p. 26): "A path drawn by the POLY procedure will close precisely when the total turning reaches a multiple of 360°." Regular polygons result when that multiple is 360°, but the theorem implies that the turtle will eventually return to the starting point for any integral input to ANGLE.

VECTORS

Vectors are ideally suited for presentation via turtle geometry due to their distance-direction nature. Geometrically, a vector may be thought of as a segment of given length pointed in a given direction with respect to some frame of reference (e.g., the Cartesian plane). A vector beings at the origin and ends at a point the given distance away from the origin in the given direction. Note that the angles related to the vectors are with respect to the positive X-axis, and not with respect to the direction of the immediately preceding segment, as is true in POLY. Therefore, vectors with positive angle measure are really generated by left (counterclockwise) turns as in trigonometry, with the turtle's initial direction horizontal to the right (toward the positive X-axis). Thus, to draw a vector of length 50 in the direction 60° would require commands of LEFT 60 FORWARD 50, assuming the turtle was initially at the origin and pointed to the right (or some equivalent set of commands).

Two operations on vectors are considered in this paper--the addition of two vectors and the multiplication of a vector by a constant. For example, LEFT 60 FORWARD 3*50 would produce a vector in the same direction but three times as long as the one produced above. On the other hand, the operation of adding two vectors is much more complicated, unless the directions are the same or opposite. If vectors are symbolized by [x,y], where x is the direction angle and y the length, then to add [60 100] to [60 300] would require only LEFT 60 FORWARD 50 FORWARD 3*50 merely adding on the additional length in the same direction.

To add two general vectors of the form [x y] and [a b] is a complicated procedure. If the turtle begins at the origin and travels a distance of y in direction x, and from that point travels a distance of b in the direction a, the sum of [x y] and [a b] is defined to be the vector with initial point at the origin and terminal point at the terminal point of the second segment drawn above. The length of the sum vector is the length of the diagonal of a parallelogram and the angle is a function of the angles of the two added vectors as well as their lengths. (See the figure below.)



Our approach has been to develop a procedure that will (1) draw the two vectors, (2) draw auxiliary segments to complete the associated parallelogram, (3) draw the diagonal of the parallelogram (the sum vector), (4) erase the auxiliary segments, and (5) print the length and angle of direction of the sum vector on the screen. Several approaches to the problem of developing a procedure to add vectors are possible, some requiring insights into coordinate geometry, trigonometry, or other mathematical topics, and all require great familiarity with the capabilities of LOGO and with task analysis. Programs developed for the addition of vectors are too lengthy to be shown in this paper, but may be obtained by contacting the authors.

REFERENCE

Abelson, Harold and diSessa, Andrea. Turtle Geometry, MIT Press, Cambridge, Massachussetts, 1980.

COMPUTER LITERACY IN A MULTICULTURAL ENVIRONMENT: PROGRAMMING OR APPLICATION PACKAGES?

Ernest Carey
Brigham Young University--Hawaii Campus
Laie, Hawaii 96762

INTRODUCTION

With most professional journals extolling the accomplishments of computers in the elementary and secondary schools, one might be tempted to ask if a computer literacy course at the university level is even relevant. It is encouraging to read of computer literacy at the elementary and secondary levels, and we can talk about the day when computers are as familiar to students as TV, but this will not happen for quite some time. The use of computers in elementary and secondary schools is still far from universal and when it finally becomes so we must wait for the "computer literate" students to filter through the system and reach the post-secondary institutions. Until that happens, Computer Literacy will remain a significant issue at Brigham Young University -- Hawaii Campus (a satelite campus of BYU--Provo) and other post-secondary institutions.

Though I currently teach the computer literacy course (Introduction to Computer Science and Data Processing) at BYU--HC, most of my formal training is in mathematics, statistics, and education. I mention this only because I feel it is fairly typical of most educators involved in computer literacy education at the pre-college level and on small post-secondary campuses. I also feel that the heavy emphasis on programming in many of these courses is largely a result of the preponderance of teachers with backgrounds similar to mine who are currently involved in teaching them.

This paper proposes that the prominent role that programming plays in today's computer literacy courses is largely due to "tradition" and that there might now be a "better" way to provide students with hands-on computer experiences.

GENESIS OF THE STATUS QUO

A brief look at the evolution of a typical computer literacy course, through the eyes of a mathematics teacher, might clarify my point. Articles started to appear in The Mathematics Teacher as early as 1958 explaining how computers might be used to improve problem solving skills by allowing students to develop and program mathematical algorithms. By the middle of the 1960s many individual math and science teachers were using computers to enrich their classes. The feeling was that any student who could tell a computer how to solve a problem must understand the underlying concept. I recall the saying "If you want to learn a concept - teach it. If you REALLY want to learn a concept - teach it to a computer."

Hardware resources were scarce for the beginning courses and teachers often had to ask for computer time from sympathetic commercial users. Some schools managed to lease limited computer time and install a terminal or two, and some even managed to buy their own time-share systems as prices dropped to within reach of educational users. The introduction of the microcomputer was viewed as the technological breakthrough that would put a computer in every classroom.

Software development has always lagged behind hardware development and education is usually behind business and industry in the utilization of new technology. For these reasons, the availability of good software has always been a problem for educational users. Math teachers have never looked at the software shortage as a real crisis since they intended to use the computer as a "tutee", a willing and patient "student" the math student could teach through programming.

As word spread that computer classes were enjoyable, the math teachers found students from all disciplines in their classes, students for whom a straight diet of mathematical programming wasn't too appealing. Curriculum developers also felt that the introductory computer courses might be broadened to include something besides programming to better reflect the widening role of computers in society. The "Computer Literacy Class" was born, but programming continued to be the only good alternative for hands-on experience with computers and it did provide an opportunity for students to more fully understand how a computer worked its miracles.

Since the introduction of microcomputers, a tremendous amount of quality, reasonably priced software has become available, but programming is still considered to be the most meaningful hands-on experience for a computer literacy class. Though arguments abound concerning the "best" introductory programming language, few argue about the value of learning SOME programming language.

WHAT LEVEL OF PROGRAMMING?

Most computer literacy classes don't teach machine language, and any mention of assembly language is cursory. We tend to concentrate on one of the high level languages, with personal preferences, machine limitations, or other non-pedagogical reasons being the prime motivator for the choice. Strangely enough, the move in the computer industry is toward the "friendly computer", the query language approach.

If a query language is in our future then we might consider the value of learning some of the "code" involved in the use of application packages. Though application packages vary in their "friendliness", all require some conforming to unique command structures which are then translated and acted upon by the computer. Might this be termed a level of programming? Might these packages prove more relevant to students in a "one-shot" computer literacy course than instruction in a currently popular high level language?

COMPUTER LITERACY AT BYU--HC

BYU--HC looks primarily to the South Pacific, Hawaii, and the Asian Rim for most of its approximately 2,000 full time students. Many of our students come to our campus with little exposure to modern technology and few have had any personal contact with computers. Because of the unusual feeder populations and the wide diversity of cultures on our campus (27 nationalities represented in the most recent semester), the needs of our students may be viewed as somewhat unique. However, we do have a fair representation of "mainland USA" students on our campus and those who attend our introductory computer course have not had much more exposure to computers than their non-mainland counterparts.

Our business division attracts more majors than any other division on campus, though our newly introduced Computer Information Systems degree (administered through the Mathematics, Natural Science & Technology Division) is gaining in popularity. Our computer literacy course is required for all majors within the business division and is the first C.S. course in the CIS degree. Many other majors on campus either require or recommend the course and the demand for the course always exceeds the limitations of the available hardware.

Our approach has been fairly traditional with approximately 2/3 of the class time devoted to the theories of C.S. and D.P. with the other 1/3 being spent on programming in the BASIC language. Pascal is the language that has been chosen as the "vehicle" for teaching programming techniques in our other C.S. courses. We have debated, as have others, whether the introductory BASIC of the computer literacy class was a benefit or a liability to those students continuing in the C.S. sequence. Thought has been given to substituting another language, such as Pascal or Logo, in the introductory course but a decision was made to go one step further and not instruct the students in the use of any high level language. Instead, we decided to try the introduction of several application packages.

CONCERNS

1) Will the non-computer related major (if there is such a thing anymore) benefit more from an introduction to application packages than a cursory introduction to programming?

2) Will the students in successive C.S. courses be restricted without an introduction to BASIC in the computer literacy course?

This year has been a year of experimentation for us. Two sections of the introductory course have been offered each semester with approximately 60 students in each section. The two sections studied the same theoretical concepts from the same text, but one group gained its hands-on experience through programming in BASIC while the other group worked with application packages.

Those in the "application" groups were introduced to five packages selected from:

1. word processing
2. spread sheet analysis
3. graphics
4. statistics & graphing
5. data management

They were required to use the word processing package for all work related to the course and were encouraged to use it for other classes. To facilitate students in this group who wanted to explore programming in BASIC, self-paced materials were made available.

Those in the "programming" groups were told about the application packages and encouraged to study the documentation and use them, but no effort was made to teach the packages to students in those groups.

All students were given the same pre-test and post-test and a two part questionnaire concerning their attitudes toward the course and computers in general.

Those students moving from the introductory course to the next course in the C.S. sequence are being monitored to check for any differences in performance that may be due to the different treatments in the computer literacy course.

CONCLUSIONS

This two-semester study will be completed on April 15th. Results and conclusions will be available by the time AEDS 83 convenes.

A MICROCOMPUTER EXAMINATION SUPPORT SYSTEM

Dr. JoAnn C. Carland
Dr. James W. Carland
Western Carolina University
Cullowhee, North Carolina 28723

ABSTRACT

A microcomputer system was developed to maintain an examination question data base. The system generates randomly selected examinations purposively modified to employ user specified mixes of topics and difficulty factors. The system was written in BASIC for a CBM 8032 and employs floppy diskette storage. Such systems have long been available on mainframe computers. The microcomputer offers security factors, cost parameters, and access features superior to those of a shared mainframe system without power loss. The menu driven, screen prompted system is user friendly permitting use by people without computer experience and eliminating the need for training or operating manuals.

Faced with large class sizes, many instructors at the university level have turned to the use of standardized examinations which can be scored by a computer test package employing an optical scanner for the reading of answer sheets. There are numerous excellent test packages available for mainframe computer systems which permit tremendous work savings. Nevertheless, preparing the standardized examination remains an ardous task, even when one maintains a body of questions utilized in the past from which to draw. There have been main frame computer packages developed to maintain banks of test questions and provide for the random generation of examinations. Such systems can be expensive to maintain and consequently have more frequently been used by departments for common or competency exams rather than by individual instructors. With the declining cost of microcomputers, it has become possible for a single instructor to prepare and maintain an examination generation system which is individualized, inexpensive and perfectly secure.

Utilizing a CBM 8032 microcomputer with a one megabyte CBM 8050 floppy disk drive, we developed an examination support system. The system was written in BASIC and designed to be menu driven and screen prompted so that operation by untrained individuals is simplified. Security is afforded not only by the requirement of a CBM system for the reading of a data diskette, but by physical protection of both data and program diskettes, and by the employment of a password for access to existing questions. The program was compiled using the DTL BASIC Compiler by Drive Technology distributed by Canadian Micro Distributors, Ltd. The compilation results in code which is not legible by any but the most sophisticated programmers, thereby affording another layer of security as well as enhancing the run time speeds of the system.

The system permits file maintenance of the examination questions through menu options for record addition, change, and deletion. For each test question the data file contains: the question, three detractors and the answer; the topic area; date the question was added to the base; number of times the question has appeared on examinations in the last 12 months; the number of students tested by this question in the last 12 months; the total number of students tested; the number of correct responses to the question in the last 12 months; and total number of correct responses to the question. These last four records permit the calculation and display of the ratio of correct responses during the last 12 months and since the addition of the question. The number of appearances of a question is maintained by the system as a by product of generating examinations. The correct response ratio is updated through a menu option for entering the results of each examination. The updating option asks for the number of students sitting for an exam and for the number of correct responses for each question.

Random generation of questions followed by the printing of an answer key is present but has been enhanced through the possibility for purposeful selection of questions. When the menu option for examination is selected, the user is asked a series of questions which are used to modify the random selection process. The user is permitted to identify the initial question universe by including the entire data base or by selecting up to five question topic areas. The user can modify or limit the universe by establishing a maximum number of appearances of examination questions as a further criterion for inclusion. Finally, the user has the ability to specify a difficulty factor for the examination. The difficulty factor takes the form of an indication of the percentage of questions desired with correct ratios of 90% to 100%; percentage of questions with correct ratios of 80% to 90%; percentage of questions with correct ratios of 70% to 80%; and so on in 10% steps down to correct ratios of 0% to 10%. The correct ratio requirements can be overall requirements or experience during the last 12 months only, as selected by the user. The final input is a request for the number of questions desired on the resulting examination and the number of questions to appear in each of the five topic areas if these were selected in the beginning.

The examination generation module will utilize the random number generator function to select a question at random and will examine whether the question fits into the universe. If a selected question does fit within the user indicated universe, the question will be compared to the previously selected questions to ensure that the difficulty level percentage requirements are not exceeded. If insufficient questions are available under the user restraints to generate the examination, the system will indicate this fact on the screen, report the number of questions selected so far in each difficulty class, allow user modification of the restrictions, and continue the process. When the desired examination has been selected, a disk file is created with the question numbers generated and the date of the examination. This file is now to be used to produce one or more printed copies of the examination. The file will be maintained until the examination results are entered into the system at which time the primary data base is updated and modified for the appearance of questions, and for correct ratios and the temporary examination file is eliminated.

The micro examination support system does not result in a loss of power or security compared to a mainframe system. It does result in more favorable support cost patterns. More importantly, the system permits the customization of examinations. Several instructors utilizing the same system are afforded the opportunity to develop and maintain individualized data bases, or to employ the same data base but prepare exams which reflect their respective teaching emphases in terms of material coverage, timing of question appearance and test question difficulty patterns. Despite the employment of a microcomputer, the level of system sophistication has not suffered. Enhanced access factors and ease of use result in better employment of the system and that translates into the consumption of less time in testing, therefore more time for teaching and research.

COMPUTING IN INDUSTRIAL EDUCATION
An Often Forgotten But Very Fertile Field

James A. Carter Jr.
Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan
Canada S7N 0W0

Most people readily associate the topic of computers in the public schools with either the math or science department. The majority of states, however, who certify teachers of computing do so in Vocational Education. Industrial Education, one of the main components of Vocational Education, is one of the most logical areas for studying computer education.

This paper will discuss the results of research into the uses and priorities of computer applications within 61 university departments offering Industrial Teacher Education programs in the midwestern U.S.. It will discuss the applications used, the priorities used, the resources available and lacking, and the effects of a number of demographic and technical factors on the uses and priorities of these applications.

Amongst the various fields within education, Industrial Education has a special concern for the study of technology and society. Based on various statements of its mission, one can expect that there should be considerable attention focused on computing within Industrial Education. This paper reports findings that confirm this expectation.

Research of related literature found eleven major applications of computers used by various Industrial Teacher Education departments: Administrative Data Processing (ADP), Word Processing (WP), Research Data Processing (RDP), Computer Assisted Instruction (CAI), Computer Managed Instruction (CMI), Computer Literacy (CLit), Computerized Instructional Design (CID), Software Technology (Soft), Hardware Technology (Hard), Computer Assisted Design (CAD), and Computer Aided Manufacturing (CAM).

A questionnaire was developed and sent to 67 members of the Mississippi Valley Industrial Teacher Education Conference, in spring 1982, to investigate the use and factors relating to the use of these and any other computer applications that were to be found. The highest ranking member of the Conference from each department represented was chosen. Responses were received from 48 (71%) of the departments. All of the uses of computing in the Industrial Education departments were reported to be in one of the eleven applications identified above. The extent of implementation by departments is reported in Table 1. A noticeable difference was observed between the percentage of departments reporting their use of various computer applications and the percentage of students using the applications. The extent of use of applications by students is reported in Table 2.

Table 1
Implementation of Applications by Departments

Application	Departments Using	
ADP	35	(73%)
WP	34	(71%)
RDP	36	(75%)
CAI	34	(71%)
CMI	20	(42%)
CLit	41	(85%)
CID	27	(56%)
Soft	40	(83%)
Hard	35	(73%)
CAD	32	(67%)
CAM	22	(46%)
N=	48	
Mean		(68%)

Table 2
Students Involved in Applications

Application	Students Involved	
ADP	500	(9.4%)
WP	289	(5.4%)
RDP	343	(6.5%)
CAI	1370	(25.8%)
CMI	632	(11.9%)
CLit	2272	(42.8%)
CID	505	(9.5%)
Soft	764	(14.4%)
Hard	1037	(19.6%)
CAD	1201	(22.6%)
CAM	1246	(23.5%)
N=	5303	
Mean		(17.4%)

It appears that some departments have just started using the various applications and have reported their use, although this use has not been passed on to their students as yet. The future plans of many of the departments include either introducing or expanding the use of computing by students.

The research found a fairly even split, 38% to 35%, in methods of implementing applications between using sources within the department and using sources outside the department, but within the university. This split was not equal for all applications. ADP and RDP, which are often centralized in the university, were used more than twice as often from sources outside the department. CAD and CAM, which would be of interest only to a limited amount of other departments, were used more than twice as often from sources within the department. In a few cases applications were reported to be implemented by both types of sources. The two main methods of implementation are reported in Table 3.

Table 3
Methods of Implementation

Application	Implementation			
	Within		Outside	
ADP	12	(25%)	25	(52%)
WP	18	(38%)	19	(40%)
RDP	12	(25%)	26	(54%)
CAI	20	(42%)	16	(33%)
CMI	9	(19%)	13	(27%)
CLit	26	(54%)	19	(40%)
CID	15	(31%)	14	(29%)
Soft	21	(44%)	22	(46%)
Hard	24	(50%)	13	(27%)
CAD	25	(52%)	8	(17%)
CAM	16	(33%)	7	(15%)
N=	48		48	
Mean=		(38%)		(35%)

The research indicates that computing activities rank high among the priorities of items to add and low among the priorities of items to cut within most of the departments.

Most applications rate a median priority of "medium" from the whole group of respondents, except CLit, CAD and CAM which rated a "high" median priority. The median priorities increased to "high" priority for four additional applications, WP, RDP, CAI, and CMI, within those departments using the applications from sources inside their department. It appears that direct experience, with at least some of these applications, increases the appreciation and priority of them.

The three main groups of lacking resources, in order of increasing frequency of need, are: human resources, computing resources, and supporting resources. The human resource of experienced faculty and staff, however, was more lacking in most applications than the computing resource of hardware.

Human resources comprise both having experienced faculty and staff to teach and support the applications and having interested students to study them. The former, on average, were twice as lacking as the latter. The extent of lack of human resources is reported in Table 4.

Table 4
Human Resources Lacking

Application	Resource Lacking	
	Faculty	Students
ADP	16 (33%)	14 (29%)
WP	13 (27%)	7 (15%)
RDP	12 (25%)	10 (21%)
CAI	13 (27%)	7 (15%)
CMI	18 (38%)	8 (17%)
CLit	7 (15%)	3 (6%)
CID	16 (33%)	6 (13%)
Soft	10 (21%)	2 (4%)
Hard	9 (19%)	6 (13%)
CAD	14 (23%)	4 (8%)
CAM	22 (46%)	5 (10%)
N=	48	48
Mean=	(28%)	(14%)

Computing resources comprise both hardware and software resources. The former is generally less lacking than the latter. The extent of lack of computing resources is reported in Table 5.

Table 5
Computing Resources Lacking

Application	Resource Lacking	
	Hardware	Software
ADP	9 (19%)	15 (31%)
WP	12 (25%)	7 (19%)
RDP	9 (19%)	10 (21%)
CAI	14 (29%)	16 (33%)
CMI	17 (35%)	22 (46%)
CLit	7 (15%)	6 (13%)
CID	14 (29%)	22 (46%)
Soft	7 (15%)	11 (23%)
Hard	13 (27%)	17 (35%)
CAD	16 (33%)	20 (42%)
CAM	26 (54%)	25 (52%)
N=	48	48
Mean=	(27%)	(33%)

Supporting resources comprise both curriculum materials and funding resources. It should be noted that the lack of acceptable curriculum materials is almost as widespread a problem as is the lack of funding. The extent of lack of supporting resources is reported in Table 6.

Table 6
Supporting Resources Lacking

Application	Resource Lacking	
	Curricula	Funding
ADP	19 (40%)	20 (42%)
WP	15 (31%)	19 (40%)
RDP	17 (35%)	20 (42%)
CAI	18 (38%)	22 (46%)
CMI	26 (54%)	23 (48%)
CLit	15 (31%)	16 (33%)
CID	22 (46%)	23 (48%)
Soft	15 (31%)	18 (38%)
Hard	18 (38%)	22 (46%)
CAD	22 (46%)	22 (46%)
CAM	26 (54%)	26 (54%)
N=	48	48
Mean=	(40%)	(44%)

A variety of demographic factors were investigated as possible influences on the levels of uses or of priorities of computer applications in the departments. These factors included:

institution type, college type, number of students, and financial support levels. The research indicates that only college type appears to have a noticeable effect. Industrial Teacher Education departments within colleges of education appear to have both higher levels of use and of priorities for most applications than do departments in other colleges.

Eleven main technical specialty areas were found to exist amongst the Industrial Teacher Education Departments involved. These technical specialties included: woods, metals, graphics, drafting, plastics, electronics, power, construction, manufacturing, crafts, and communications. The research indicates that neither any of these specialty areas nor the number of specialty areas present in a department appears to have a general effect on the levels of use or of priorities of computer applications. All of the technical specialty areas appear, however, to have associations with and potential influences on some specific computer applications. Further research is required in this area to determine the possible causes of these apparent associations.

The research indicates that most of the Industrial Teacher Education departments involved had plans for expanding their computer related activities in the following year. These plans, along with the relatively high priorities for adding various computing activities, indicate that computing activities have become widely accepted as one of the more important technical content areas within Industrial Teacher Education.

Of the sixty-four future activities, forty-eight could be categorized as to types of activities. The largest category of future activities were identified as development oriented. Information on the types of future activities is reported in Table 7.

Table 7
Future Activities Planned

Type	Occurrences
Development	19
New Course Offering	8
Modify Existing Course	8
Staff Training	7
Purchasing	6
Other/Unspecified	16
N=	64

Of the sixty-six future applications reported, fifty-six could be categorized as to types of application. The largest number of future applications dealt with CAD. Information on the types of future applications is reported in Table 8.

Table 8
Planned Future Applications

Application	Occurrences
ADP	5
WP	4
RDP	2
CAI	4
CMI	0
CLit	6
CID	3
Soft	5
Hard	6
CAD	15
CAM	4
Other/Unspecified	12
N=	66

In conclusion, one can see that computing is indeed a very fertile field within Industrial Education.

FOREIGN LANGUAGE COMPUTER LAB

Allen Clancy
Lincoln High School
1600 S. W. Salmon Street
Portland, Oregon   97205

ABSTRACT

In order to combat the isolating effect of computers; to promote a competitive spirit among students and to direct computer activity toward classroom goals, we wrote a foreign language program. With able assistance from computer literate students we developed a shell program that reads and a file which is changed each week as requested by the language teacher. The main program was written at the end of a popular game which occupies faster students and is the carrot reward. The most used file format prints an English sentence which the students translate into French, Spanish, or German paying careful attention to accents, spelling, genders, and punctuation. We use six cassettes to load twenty-four computers in ten minutes and the entire class participates. Marked improvement on written tests delights teachers; the dynamic computer room atmosphere stimulates students.

BODY OF PAPER

In August 1981, Lincoln was chosen by a midwest computer outfit as a site for its promotional plan to sell programming/literacy to the Lincoln community. Under the plan, students would have use of 24 16K Pet cassette-storage micro-computers by day as a trade-off for the use of the building by night. The effort failed and the district, with considerable community contribution, bought the computers at a discount. It was all very sudden.

Programming had, traditionally, been a semester course open to upper division math students. I continued to offer programming with geometry as a requirement so I could cover the major topics of a two-semester college course in one semester with an accelerated course offering. I have two classes and can teach programming to one hundred students a year. That leaves the computers idle four periods a day for other uses - namely computer assisted instruction (CAI).

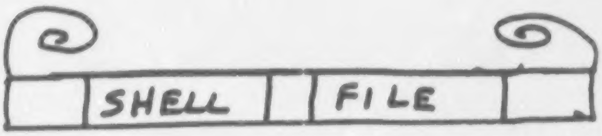
I believed what I read about CAI - of its inevitably increasing role as teacher-substitute for drill mastery. My successful use of computers in this role was not as a substitute. We tried sending small groups out of the classroom but we discovered students seldom work at programs when their teacher was absent and when the programmed material was not directly relevant to classroom activity. That's all I had - a few irrelevant programs. Teachers stopped sending students. I was discouraged. I could understand why the teachers were not enthusiastic about this plan. I like my class to be together.

I rely heavily on the group as motivation to slower, lazier students to keep up. When students are gone, it is disruptive of the unit. I give individual attention to 30 students at once for 40 straight minutes - 5 times a day. I think most of the teachers in my building operate similarly. If I take five minutes out of class time to explain something to a student who missed the day before, that's (29x5) minutes of student-time lost to the rest of the class. At first, we did try to send small groups of students out of English or Math classes to practice at the computers.

I looked for software. I sent catalogues around. I decided to invite teachers to commission drill and practice material for my programming students as one of their class projects. That's a good idea and I still use it, but by the time the students finished writing the programs, it was too late. The teachers were grateful, but mostly embarrassed. Everyone thought it was hopeless. The language teachers felt computers were going the way of the language labs.

It was with foreign language that I focused on a new plan. We have an exceptional foreign language department at Lincoln. The members are enthusiastic about new ways of improving language skills, but they needed something that pertained to current problems, especially in translation. Prepared software is never right because it is hard to predict problems. Purchasing $200 of software is seldom addressed to the needs of the curriculum.

I finally asked an exceptional student, Stuart Williams, to write a shell program with a changeable file.



The program is loaded into the computer memory via ordinary audio-cassette player. The program is run (type RUN and press return). The tape winds some more and the file is read into the program shell. The program, itself, never changes from language to language. The program shell asks 20 separate questions and compares twenty answers. The changeable file contains the secrets. The file is written behind the main program. I make six copies and load twenty-four computers in ten minutes. The program typically displays an English sentence and compares the student's typed-in translation with the answer string - also read from the file. If the student's response doesn't compare exactly - letter for letter, period for period, the student gets a negative response from the computer along with the correct answer - the same answer string that was read from the file.

The questions and answers are represented by a doubly subscripted string variable and are chosen by random selection. On the foreign language programs we have, normally, twenty questions. The student must get forty correct to finish. The students will get many of the questions more than once. "Repetition is the mother of learning" to quote a Russian saying.

I asked a Spanish teacher to write down samples of the translations for an upcoming test. She gave them to me and the next day I was ready to invite the class to practice translations. The students enjoyed the experience and the teacher enjoyed watching the students concentrate and be precise enough to earn a positive response from the computer. Since that first success the French, Spanish and German teachers have brought their classes to the computer center every week.

Being able to prepare a computer lesson at a day's notice is what makes the computer lab work. Having the computers located in a single location is also important. Fifteen computers can work as well because students like to work in pairs.

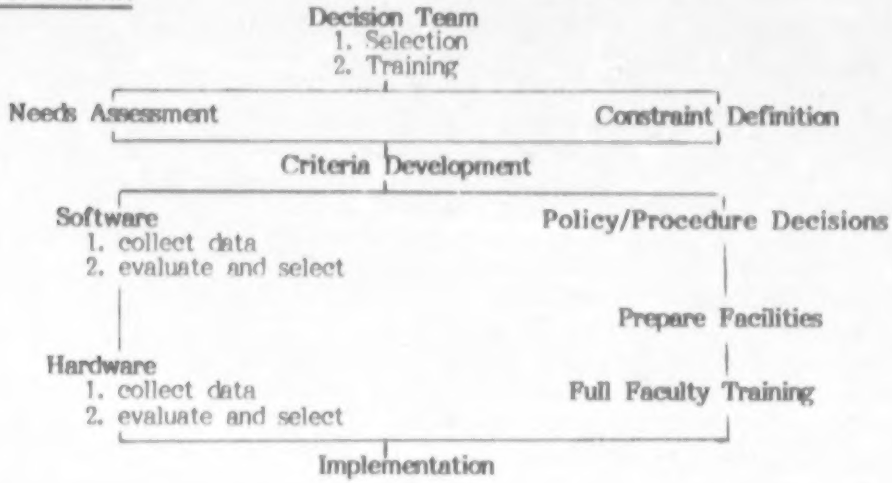
Selecting Micros Is A Sequential, Not A Random, Activity

Frank W. Connolly
The American University
Massachusetts at Nebraska Avenues
Washington, D.C., 20016

ABSTRACT

Too often schools make decisions regarding computers in the wrong sequence. This paper presents a model for planning the selection and implementation of microcomputers in schools. The model focuses on the proper sequencing of the decision steps.

BODY OF PAPER



Outline of Model for Planning the Selection of Microcomputers in Schools

When it comes to selecting, installing and implementing computers in an educational environment, regretfully, many organizations make decisions in the wrong sequence.

A quick survey of schools in the Washington, D.C. area brought almost universal agreement on the critical factors:

- o hardware
- o software
- o costs
- o vendor
- o support

With the exception of one or two schools, most considered these when making the purchase decisions for microcomputers. However, they didn't realize that the sequence of considering the factors was important.

When not done in the proper sequence, schools fail to gain the full benefit of their computer dollars. They end up with hardware and software that is not properly matched to their specific needs. In these days of tight budgets it makes sense to maximize the benefits for every dollar in your budget. The most frequent and costly mistake is to start by considering hardware. Due to its complexity, and the marketing efforts of numerous vendors and manufacturers, educators are of the mistaken impression that hardware must be selected before anything else. Don't get me wrong . . . it's important. But, other factors should be considered first. In fact, the hardware decision is not as complex or difficult to make, when done in the proper sequence. The first few steps prepare your and your school to make computer-related decisions.

The first step is to select and train those who will do the planning and make decisions regarding your computers. From there, the process is driven by how you and your staff intend to use computers in your school. After the uses are defined, the software to meet those needs is identified, and finally, the hardware to handle the identified software is selected.

Several steps in the proposed model are common to many planning models, several are unique. The remainder of this paper will concentrate on defining and explaining the unique aspects of this model.

Decision Team

Large school systems apparently have an advantage in planning — they can afford to hire and pay specialists to do it. Small systems, and independent schools are at an apparent disadvantage, as they must rely on their existing faculty and staff to plan. Such an arrangement may indeed have a negative impact under most circumstances, as it places added burdens and responsibilities on personnel resources. But, when it comes to making computer decisions, the added responsibility is an advantage. To be integrated successfully into your curricula computers must fit easily into the classroom setting. Teachers must be comfortable with the machines, and comfortable with how they are being used in their classes. Your chances of reaping the maximum benefit from computers is higher when the faculty seek the resource, rather than the principal, or administration order its implementation. When an autocratic administrator or a centralized staff studies computers and makes system-wide decisions on their implementation, the one most impacted — the classroom teacher — is not involved. The result is too frequently disappointing. It may be the most expensive system involving state-of-the-art resources, but without acceptance and enthusiastic support in the classroom it fails to fulfill its potential. Therefore, getting teachers and staff involved is critical.

Computers are extremely flexible. They can be effectively integrated into almost any course, subject and grade. They can also be effectively utilized in staff and administrative functions in your school. This means that decisions made about the selection and installation of computers can affect every member of your professional staff. This flexibility is both a blessing and a burden — a blessing because so much can be done with them, once they're in, up and working — a burden because the process of selection is complex. The decisions should not be made from one person's perspective. The interests and concerns of all parties affected need to be considered, expressed and involved from the beginning. It's too late once the order is placed, or after some of the philosophical issues have been resolved, or once you're into the selection process.

Everyone's input and ideas are needed, but decisions can never be reached by large groups of people — too many opinions, too many personalities, too many distractions and disruptions. So?

Establish a representative team to plan for computers, evaluate your alternatives and make the decisions.

The most important factor to consider in selecting members for the team is their awareness and understanding of your school — its teaching, pedagogy, learning objectives, its goals, internal politics, student population, and the individual faculty and staff members. These factors are unquestionably more important than an understanding of computers. It takes years to develop an effective teacher; you can learn enough about computers to make good decisions in a few days.

Based on this premise, involving "volunteer experts" is to be avoided. Volunteer experts are interested and well-intentioned parties who by virtue of training and experience attempt to assist you in grappling with the complex issues of computers — parents who work with and at computers, or representatives from software houses and computer stores, or involved volunteers who offer their computer expertice. Their motives are good, but because you want to integrate computers into an existing environment, their assistance is usually not helpful. It is knowledge of the environment that is crucial to good decisions.

Limit your team to five to seven members, provided each can realistically represent a specific area — upper grades, primary, science departments, staff, administration. Although not necessary, it is recommended that the administrative representative have sufficient authority to committ the resources of your school once a decision is reached. As the administrator will have to learn all about the computer anyway, he/she should have the benefit of the multiple views and problems the team will identify.

The second step in preparing a Decision Team seems obvious when we're examining this process from afar, but it is frequently forgotten in the heat of making decisions. It is a truism that lies at the foundation of our profession -- no individual or group can be expected to make reasonable decisions regarding an issue if they do not understand the issue or its implications. Which is better suited for elementary grades, LOGO or BASIC? Which has better cost efficiency for your school cassette tape or disk? Is the pedagogy of MagicSpells acceptable for a particular grade, or class? What about PASCAL for high school classes? What kinds of resources are needed to teach Computer Literacy in your kindergarten? Third grade? Junior high? Only when you understand the appropriate concepts, and terminology, can you make reasonable judgements about the computer.

This isn't to say you need a degree in computer science or years of experience in the computer industry to select computers for your school. It does mean that those who make up your selection team need to get some training **before** they attempt to grapple with the selection process. The training does not have to be exhaustive, but it should be objective, should involve "hands-on" experience, and ideally be oriented to educational institutions. Twenty or thirty hours of training on microcomputers in business is less helpful than ten or twelve hours on computers in schools. Business-oriented courses won't touch such vital topics as using drill and practice routines, or resources available for tailoring packages to indivually paced environments, or the reward structure or re-enforcement that should be evaluated for educational software.

The reason for objective training is to avoid bias. Local computer stores offer courses advertised as "general" in nature. However, they have ulterior motives. The same is true for software houses. Therefore, in researching introductory courses, be sure that both the sponsoring organization and the instructor are not involved in a conflict of interests. If they are, you won't get the complete picture you need to properly evaluate your alternatives.

One of the greatest benefits of a computer course is derived from "hands on" experience. No matter how entertaining the instructor, computers are a skill-oriented topic. You learn best about computers in a structured environment, where the teacher is a guide or coach, not a lecturer.

Needs Assessment

Once you understand the power and limitations of computers, you can begin to address the question of what you want the computer to do in your school. While this evaluation needs to be done with as much precision as possible, don't loose sight of the fact that computers are going to be absorbed into existing classes. They are one tool in your educational tool box, they are not intended to dominate the educational program.

Your Needs Assessment has to be specific. Table I includes an abbreviated list of topics that require answers. If you don't know all the answers, find out, or take the time to prepare a reasonable estimate.

Constraint Definition

This phase of the planning process can take place at the same time as the Needs Assessment, although it is not a requirement. This phase involves defining specific factors that limit your choices. By defining realistic boundaries, you eliminate irrelevant alternatives, saving the time and energy required to evaluate them.

Table I

Subjects
    grade level
    lesson(s)

Pedagogy
    simulation
    drill & practice

I/O Requirements
    color of b/w
    keyboard
        upper/lower case
        scientific/language notation

Access
    centralized or in each room
    supervised or open

Classroom
    work individually or in groups
    individually-paced or lock-step

Be realistic. How much money can you spend on equipment purchases? What are the budget limits for maintaining and operating the equipment in the future? What is the likelihood that a faculty position can be added just for doing the computer-related coursework? Where will the equipment be housed, and how much money is available for modifications? How much support can be expected from the Board? Parents? Faculty? Staff? Principal?

Criteria Development

After Needs Assessment and Constraint Definition are completed, the team is ready to develop a statement of the criteria that will be used in evaluating various alternatives. The Criteria are a statement of requirements based on the identified needs, reflecting the constraints. For example, if there is a need for color output on the monitor, then the criteria will require that any system selected include a color monitor. Once again, the object is to have the statement in specific terms, and in writing. Specific terms will eliminate much of the subjective guesswork that tends to creep into the selection process. Having it in writing will require team members to resolve issues and accept compromises while focusing on the internal needs and requirements of your school.

All requirements are not of equal importance, so, factors should be prioritized. At a minimum, identify those items which are mandatory, and those that are desirable. If a requirement is defined as Mandatory, then an alternative that does not include the requirement should not be considered. Table II offers some specific factors to consider in your criteria.

At this point you have a precise statement of your needs and the criteria to use in evaluating your options. Both of these statements should be written in general terms and should not identify a particular package or brand. If your Decision Team has done its work properly, the hard part is over.

Table II

- Application-oriented
  - language requirements
  - pedagogical approach
  - keyboard (notation, etc.)
  - monitor (color, b/w)
- Budget-oriented
  - initial cost
  - maintenance
- Faculty/Staff-oriented
  - documentation
  - friendly software
- Vendor-oriented
  - location
  - repair & maintenance services

Software

Software is more critical to you than hardware. It is the software you'll be involved with. It will seek information and place limitations on what you and your students do. Therefore, chose software first, then determine what hardware is needed to support it.

The first step is to collect data. Go to computer stores; talk with the sales people; visit other schools and see what they are using and how they like it; call in your parents group for suggestions; read periodicals, books, directories. Don't forget, you're interested in software, not hardware. Your investigations are to focus on what software packages fill your defined needs, and how current users feel about them. In evaluating opinions, consider the source. The professional evaluation of an educator using the program in an actual class setting is far more valuable than the opinion of a computer expert. Marketing claims, and views of sales people and vendors need to be recognized as sales statements.

Next, take the collected data and match what you've learned about software against the Mandatory Requirements of your needs statement. Any alternative that doesn't fulfill your Manadatory Requirements should be eliminated. Whatever software packages are left should be evaluated according to the Desirable Requirements you identified. The result will be a shopping list of software, ranked according to your priorities. Now you can move on to the choice of hardware.

Hardware

Once again, you must go back to visiting other schools, and stores, and parents, and "volunteer experts." User groups can also be a helpful resource at this point. Members of such groups are familiar with the hardware and its problems, and have experience with the various vendors, and can give you data on training, supplies, realistic support levels, and a myriad of other factors. This time, your inquiries are even more focused than before. Now, you want to know what hardware supports the software you have chosen.

You'll find there are a limited number of hardware alternatives open to you. Armed with your hardware information, the Team can make the hardware selection.

Policy/Procedure Decisions

This is just the beginning. Having and using computers in your school will place continuing requirements on you and your faculty. Many of these need to be addressed before the system is installed. The temptation is to put off establishing operating procedures until a problem arises. However, it takes significantly more time to recover from problems than to avoid them. Table III provides a sample of potential areas that should be addressed.

Table III

- Software-oriented
  - documentation standards
  - sharing programs & data
  - file backup
- Access
  - supervision requirements
  - physical security
  - class scheduling
- Budget allocations
  - additional software/hardware
  - periodicals
  - user group membership
  - professional meetings

SUMMARY

Ready or not, computers are here. As more and more independent schools and systems enter the computer age, they have to make difficult decisions regarding computers. Too often the decision steps are taken in the wrong sequence. This paper outlines a model for the decision steps needed to make the initial selection of software and hardware. Considering factors in the proper sequence, can save time, money and frustration. It leads to decisions that are more readily accepted, and to installed systems that are more fully utilized.

ADVENTURES IN EDUCATION
Educational uses for computerized adventure games

John B. Cook
Area Education Agency 6
210 South 12th Avenue
Marshalltown, Iowa 50158

ABSTRACT

Adventure games designed for entertainment are being used in schools to achieve educational goals. Other adventure games are written primarily for education. An example is Evolve, by the author. Students also learn when they create their own adventure games.

Today's youngsters may find themselves going on a mission in outer space, fighting a life or death battle with the mythical minotaur, interviewing suspects to solve a crime, exploring ancient temples, fleeing from Tyrannosaurus rex, or piloting a miniaturized submarine through the blood stream of an accident victim to perform life-saving brain surgery. All this is possible through computerized adventure games.

CHARACTERISTICS

The list of adventure games available is extensive, varied and growing. In spite of their differences, they share some general characterisitics. First there is an element of fantasy. Players take roles that foster personal involvement. Second, they must move through a representation of three-dimensional space. (Some of the less imaginative games are planar.) The players have choices of directions at intersections in a maze, cave, castle, forest paths, or whatever the imaginary terrain happens to be. Third, the programs are primarily command driven and usually not in real time. The players must continually make considered choices that affect the outcome of the game. Since the games are not in real time, the players can consult data, notes and maps, and give extended thought to tough logical puzzles. An adventure game player is as different from an arcade game player as a physicist plotting the trajectory of a rocket is from a baseball player catching a fly ball.

EDUCATIONAL USES

The fantasy and active involvement are intensely motivating. Players spend hours in intense concentration, sometimes pursuing a goal for months. It is no wonder that educators are looking for ways to harness this motivational power for educational purposes.

Adventure games written primarily to provide entertainment are being used for educational purposes at home and, often informally after normal school hours, in schools. Students using these games must apply several important skills:

Mapping is a specific skill that students must develop and polish to become successful adventurers. This skill may not be a part of the traditional school curriculum, but it is an important survival skill in our mobile society. A significant number of people lack this skill, and it is a real detriment to their employability, independence and self concept.

A significant amount of reading is involved in playing adventure games, especially the all-text games. For some students, the game may provide a motivation to read that is otherwise lacking. In any case, it provides practice in reading for comprehension and vocabulary growth. The games may even be considered worthwhile as a kind of electronic literature. They have been called the novels of the future. While some games may be pulp fiction, others may be modern classics.

Informal spelling practice is provided as students type commands and names of objects. The computer does not tolerate incorrect spelling.

A great deal of practice is provided in logical problem solving. This is recognized as a primary educational goal. Students often make use of such specific problem solving aids as keeping records and notes and consulting manuals.

Finally, adventure games provide computer literacy activities including reading instructions on a screen and using a command-driven program.

EXAMPLES

Simple adventure games have been around for a long time. Some examples are Hunt the Wumpus and The Devil's Dungeon. However, the grandaddy of the present generation of full-blown, sophisticated adventures is the Colossal Cave Adventure by Don Woods and Willie Crowther of M.I.T. Their original adventure has been modified and transported to many existing computers and is available from both commercial and noncommercial sources. Some commercial adventures that have been mentioned as having educational value include Odyssey by Robert Clardy (Synergistic Software), Ultima by Lord British (California Pacific), and Deadline by Marc Blank and David Lebling (Infocom).[1] The Zork series, (by Blank and Lebling, Infocom) has been promoted for educational use by the Zork Users Group.[2]

One of the more widely used Adventures for the Apple computer is The Wonderful World of Eamon.[3,4] This is a series of adventures that are all in the public domain. The series allows the user to take the same character through an unlimited number of different adventures. The master diskette and several of the adventures were written by Donald Brown. There are now more than thirty Eamon adventures written by a variety of authors. The Eamon adventures are available from A.P.P.L.E., among other sources, for about the cost of a blank diskette.

PROBLEMS

There are several problems with using adventure games in school. One of these is time. Most adventure games take several hours to play, and some take much longer. It is essential that students be able to save their games at the end of a class period and return to them later. Even then, it may be difficult to justify taking that much time from a curriculum crowded with what are considered to be essential skills.

Some of the games are too difficult for younger or less able students. The very students who might benefit most from the motivational quality of the games may be turned off through frustration.

The content of games written purely for entertainment may not match the educational goals of the school. It may be difficult to see the real world value of killing dragons with magic spells. Some of the more violent adventures may be teaching antisocial behavior.

ADVENTURES WRITTEN FOR EDUCATION

Some of these problems are being solved in adventure games written recently that are intended for educational use. These educational adventure games may be seen as an extension of simulation games, like Oregon Trail, that have already been accepted for their educational value.

One such educational adventure game is Evolve, by the author. Evolve is an adventure in evolution. It takes the user from the "big bang," through the evolution of life on earth, up to the present day. The motivating power of computer adventure games is used to teach concepts of evolution. The principal objective is that the student will learn the basic outline of the evolution of life on earth. Secondary objectives include mapping skills, computer literacy, and developing a more positive attitude toward science through a motivating and entertaining experience. In Evolve the user follows the branching progress of evolution by moving through a series of stations. At each station the user "becomes" an organism (or, in the early stages, non-living matter). Through trial and error, the user must choose a route that leads to the present day, preferably ending up as a human being.

Another sophisticated educational adventure game is Microbe: The Anatomical Adventure, by Robert Clardy and Alan Zalta. The scenario is that of Fantastic Voyage by Isaac Asimov. A miniaturized submarine must be piloted through the body's circulatory system in the brain of an accident victim. Microbe presents information on the human body, immunology, and safety topics. The difficulty levels may be varied to suit a wide range of users. It may be used by individuals, or roles may be assigned for group play.

CREATING ADVENTURE GAMES

Higher level educational objectives can be achieved by having students create their own adventure games. Students can be motivated to write, organize and synthesize at levels far beyond the typical school writing assignment. A useful tool in this effort is the public domain Eamon Dungeon Designer's Diskette for Donald Brown's Eamon series. After planning the adventure, the students simply enter the room descriptions, connections between rooms, "monster" characteristics and descriptions, and artifact descriptions. Programming ability is not necessary, but students with some knowledge of programming in BASIC can spice up the adventure with special effects. A "Dungeon Designer's Manual" is provided that may be printed from the diskette.

CONCLUSION

We are at the frontier of educational use of adventure games. Short, easy adventure games that address specific educational objectives can be powerful tools for learning. An authoring system like that of the Eamon Dungeon Designer, but less combat-oriented, could help increase the supply of educational adventure games and provide a rich creative experience for learners.

REFERENCES

1. Richards, John, "Why I Let My Kids Play Adventure Games," Classroom Computer News, January/February 1983, p. 73.

2. InvisiClues, Milwaukee: Zork Users Group, 1982, p. 25.

3. Brown, Donald, "The Wonderful World of Eamon," Recreational Computing, July/August, 1980, p. 33 ff.

4. Plamondon, Robert, "An Adventure Game for the Apple II With (Almost) Everything," Creative Computing, January, 1983, p. 94 ff.

AN ATTENDANCE SYSTEM FOR SECONDARY SCHOOLS

Wilbur C. Dorsey, Division Director
Data Processing and Pupil Accounting
Wichita Public Schools
432 W. Third
Wichita, Kansas 67203

ABSTRACT

The pupil, his parents, the school staff,and district staff, demand accounting of attendance in the schools. This on-line attendance system provides the ability to record absences and classify the reason for absence in 25 catagories. On-line a pupil's record can be viewed for the day or for the entire year-to-date. Many reports can be made from the data collected hourly via a light pen marking absent pupils when the class is displayed on the CRT. Flexibility allows the principal of each school to establish his own operational procedures for attendance accounting.

Introduction

Each day, a school office receives many calls from pupils and parents regarding attendance at the school. The school office wants to know if a pupil is present, tardy or absent in each scheduled class throughout the day.

Beginning in 1973, an optic mark sheet was used to record attendance in a class each day, with the sheet being scanned on a weekly basis. As this method caused the record to remain with the teacher and not the school office, each school devised their own "paper" method of knowing each period who was absent. Teachers were thus required to keep attendance twice; (1) the optic scan sheet and (2) the "paper" method for the office (and also perhaps for themselves).

During this time period, (1973-1981) principals were also required to notify the district attendance office if a pupil had accumulated 30 or more "unexcused" absences in courses at their school. Upon receipt of this notice, the district attendance office notified parents and the juvenile court by sending a letter about the pupil's absences. Principals received a weekly report of pupils who had been absent a total of 30 hours or more to date during the current semester.

Each school office had to keep track of unexcused hourly absences as the scan sheet did not differentiate between excused or unexcused absences. Record of the pupils reported to the district attendance office was not maintained and many pupils were reported repeatedly. Consequently, parents and the juvenile court were sent duplicate letters.

In 1979, all secondary schools were using terminals to input data, class schedules, etc. Principals asked our Data Processing Priority Committee to approve an attendance system project to give them the kind of service they desired and eliminate the dual reporting system. Two schools were used as test sites beginning in the fall of 1980. All secondary schools have used the system beginning with the 1981-82 school year.

Equipment and Software

The CPU is an IBM 4341-II, four meg memory. The on-line system utilizes CICS/VSE, leased lines @ 4800 and 9600 baud, Racal-Milgo modems, CCI-CC8 controller, Telex 276/278 terminals, light pens, and 286/281 printers. The source of pupil classes is found in a DL/1 data base, extracting the information to a VSAM file. A Telex 8020 tape drive is used in passing parent/pupil information to a PDP 11/44 for writing the parents letter on a Diablo 630 printer using the Word 11 word processing package. A list of pupils reported to the district attendance office (also for juvenile court) is printed on a IBM 3203-5 printer as are other district reports.

The Attendance System

As each school has a different physical arrangement, number of pupils, and number of staff, the building principal has the responsibility to determine the definitions used in the system, i.e., (1) an "unexcused absence, (2) an "excused" absence, (3) a "tardy" (4) a school-related activity, and others.

Procedural policies are also the princpal's responsibility. Areas such as where the pupil goes when arriving late to school, tardies between classes, when does being tardy become an absence, etc. It was found that most schools did not have written procedures for the pupil or staff to follow.

Teacher Input

Each Saturday, class rosters are printed on vertically perforated sheets. Each day of the week, the course/section number, teacher name, room and hour are printed at the top of the alpha listing of pupils. The teacher tears off the strip for the day, circles those absent with a red marking pen/pencil and follows the principal's direction for marking tardies, correction of errors, etc. The attendance is recorded in the data base, by class, and is used for all reports that have attendance data in them.

Office Input

Each Monday the school office receives class rosters for each teacher. These class lists are placed in the teacher's mail box for their input (see above). Pupil office proctors collect the hourly slips from each teacher. The proctor, using the office sign-on, keys the course and section number for a display of the class roster. Using a light pen, the absent or tardy pupil is marked with the light pen. After marking all such pupils, the "enter" key is depressed to record the data. The next course/section number is entered for the next class, etc.

Each school may choose the original absence to be recorded as "excused" or "unexcused". The attendance clerk has a program to alter the record. Absence for a school event such as athletics, field trips, etc., may be recorded by placing an "S" in the appropriate hour(s). Tardies and school events are not counted as absences when reporting attendance.

All attendance is temporarily stored in a VSAM file during the day. Changes are made to this file only during that day. At night, this file is posted to two data base segments: (1) the individual class segment where absence occurred (absences only) and (2) an attendance segment which accumulates yearly attendance by using one line per day, showing the hourly absences (all marks, X, U, S, T, etc.). If a change needs to be made after the day of absence, both records should be changed.

Reports

1. At any time during the school day, the school may print a listing of all pupils who have been recorded absent. The listing shows the pupil name, address, home phone, mother's name and work telephone, father's name and work telephone, followed by the hours absent. This listing is normally printed after the first or second hours absences are recorded so calls to parents can be made. At the end of the day, the listing is again printed for a record of all pupils who were absent on a given day.

2. A printing of the data base attendance segment can be made at anytime. It will not show the current days absences. This report has many uses, especially with parent conferences.

3. A weekly attendance report is sent to the school which shows any pupil with any absence the current semester. The listing shows the pupil name and home phone, followed by the number of absences each hour and the total absences for the semester. Each period shows "X" excused, "U" unexcused and "T" tardies. Schools are able to use any other alpha character (except "S") to indicate excused absences. The exact reason for absence is then recorded in the data base attendance segment.

4. A report for the week only, lists all pupils absent during the past week. The information is the same as report in report 1, except only the total number of absences by hour is shown.

5. Pupils under 16 years of age, who have reached a total of 30 or more cumulative unexcused absences during the week are printed on a report for each school. The pupil record is "flagged" to omit him from this list during the remainder of the current semester.

6. A report of all pupils under 16 years of age in the system who have reached a total of 30 or more cumulative unexcused absences is prepared for the district attendance office and the juvenile court. a pupil will be reported to the court only once each semester.

7. A tape record is made which contains the parent(s) name, address, pupil name and total unexcused absences to date. The tape is entered into a file for word processing to write a personal letter to the parents.

Backup

A school of 1300 pupils will require between 15 to 20 minutes per hour to light pen absences. Our average absenteeism is 10%. If, for any reason, the school is unable to complete the process that day, the teacher absence slips may be brought to the Data Center where either light pen or keypunch will be used to input the data for record updating. Equipment or line failure may cause the backup procedure to be used.

Summary

The ability for the individual secondary school to have their own procedures for attendance and to be able to inquire at any time the status of a pupil has been a welcome relief to the school staff.

The entire system continues to rely on the accuracy of reporting by the classroom teacher. The reports available clearly show the individual teacher who has failed to report a pupil who was absent all day. Reports also highlight the class "skipped" by the pupil.

Parents and pupils have shown amazement at the accuracy and complete recording of pupil attendance. It is the objective of this system to assist the schools in their accountability of pupils.

PROBLEM-SOLVING COURSEWARE FOR ADVANCED MATHEMATICS CLASSES

by Lois Edwards
Courseware Designer

Minnesota Educational Computing Consortium
2520 Broadway Drive
St. Paul, Minnesota 55113

Computers have long been used in advanced mathematics classes to incorporate programming skills into the process of learning mathematical concepts. MECC's Mathematics Volume 4 - Advanced is a package of thirteen problem-solving tools that enable students without programming ability to explore advanced mathematical concepts. Programs in both calculus and finite mathematics permit graphing, approximations, and matrix manipulations. Using these programs, students can independently explore realistic, complex problems. The computer performs the arithmetical calculations so that students can concentrate on setting up the problems and interpreting the results.

Introduction

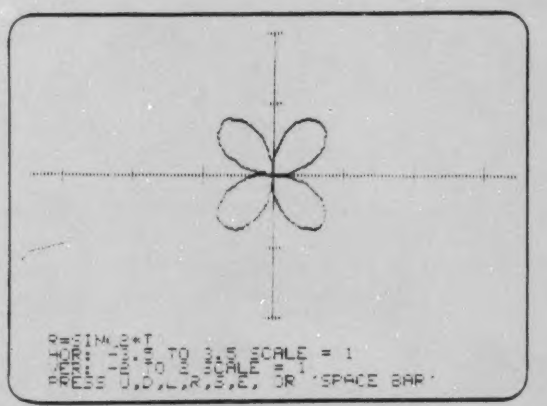
Computers have been extensively used in advanced mathematics courses to teach programming. Now courseware is becoming available which allows students to use computers in mathematical applications without needing to write programs.

Mathematics Volume 4-Advanced from the Minnesota Educational Computing Consortium provides a collection of thirteen problem-solving tools for use in advanced mathematics classes. This courseware was designed by Beverly Durkee, Augsburg College, Minneapolis, and does NOT require programming ability on the part of the students. Students simply follow instructions displayed on the screen to enter the desired equations or parameters.

Appropriate for both advanced high school and college level mathematics courses, the programs fall into two general categories: calculus/numerical analysis and finite mathematics. Rather than teaching concepts or providing practice in mathematical processes, these programs are tools that help students to study and analyze a variety of mathematical problems. Extensive support materials, including instructions for using each program and samples of typical input and output, enable students to use this courseware easily and independently.

In advanced mathmatics courses, it is assumed that students have had sufficient academic training to learn from the lecture-demonstration method in conjunction with independent textbook study. Therefore, traditional tutorials and drills have limited value in these situations. New forms of highly interactive tutorials, however, do hold promise in helping students gain an intuitive understanding of the underlying concepts of mathematics.

GRAPH



```
R=SIN(2*T
HOR: -3.5 TO 3.5 SCALE = 1
VER: -3 TO 3 SCALE = 1
PRESS U,D,L,R,S,E, OR 'SPACE BAR'
```

Overview of Courseware Objectives

Courseware can be used in higher level mathematics courses to meet several important objectives other than to instruct through tutorials or to provide practice. The programs on Mathematics Volume 4 - Advanced address these objectives.

1. to display graphs (demonstration)
2. to perform computational procedures (problem-solving tool)
    a. to solve problems
    b. to learn algorithms;
3. to emphasize numerical techniques in mathematics;
4. to study realistic applications;
5. to experiment with various approaches to a problem;
6. to manipulate variables and observe results.

Display of Graphs

A versatile graphing utility, GRAPH plots functions and conics in Cartesian and polar coordinates and also plots parametric equations. Several graphs can be plotted on the same axes and parameters can be altered as desired. Valuable as a demonstration aid in class, this program also helps individual students relate equations and their graphs. It can be used effectively in conjunction with other programs on the diskette such as POLYROOT or INTEGRATION to visualize the functions being studied.

Performance of Computational Procedures

a. To solve problems

The program POLYROOT calculates roots of polynomial functions. Two other programs find roots of various functions using Newton's method. One of these, NEWTON 1, shows each successive approximation in the process, enabling the student to experience the limiting procedure involved in the method. The other simply displays the final value of the root.

Two integration programs, one for single and the other for multiple integrals, approximate definite integrals. They allow students to solve integration problems in which no antiderivative of the function is known.

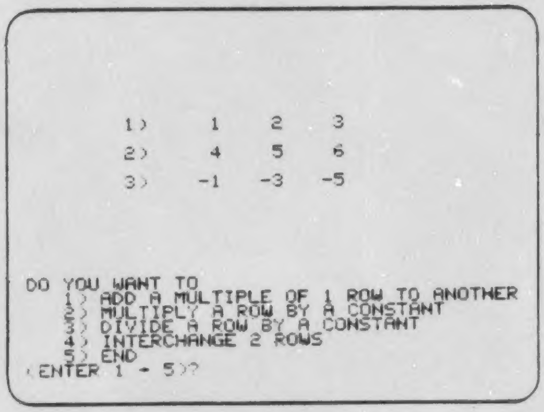
The program MATRIX quickly performs a variety of operations on matrices, including multiplying, inverting, row reducing, and finding the determinant. In problems involving systems of linear equations, students can concentrate on setting up the system of equations and interpreting the solution while the computer performs the calculations.

b. To learn algorithms

The program ROW REDUCTION enables students to learn row operations and practice reducing matrices without the tedium and frustration the arithmetic creates. The student enters the matrix elements, then specifies the desired row operation. The computer performs the arithmetic and displays the resulting new matrix. Through the experience, students learn the effects of various row operations without becoming discouraged by arithmetic errors.

Similarly the algorithm for solving linear programming problems using the simplex method is taught in LINEAR PROGRAM 1. A companion program solves the problem automatically using the same method.

ROW REDUCTION



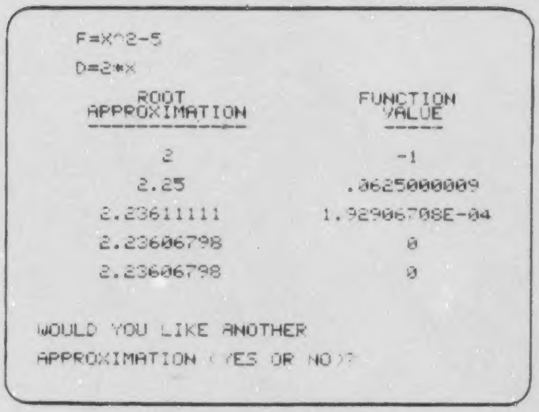
```
        1)     1     2     3
        2)     4     5     6
        3)    -1    -3    -5



DO YOU WANT TO
   1) ADD A MULTIPLE OF 1 ROW TO ANOTHER
   2) MULTIPLY A ROW BY A CONSTANT
   3) DIVIDE A ROW BY A CONSTANT
   4) INTERCHANGE 2 ROWS
   5) END
(ENTER 1 - 5)?
```

Emphasis on Numerical Techniques

The three programs that calculate roots of equations, POLYROOT, NEWTON 1, and NEWTON 2, are excellent examples of iterative techniques in numerical analysis.

The two integration packages demonstrate a variety of computational methods to find an approximation of a definite integral.

NEWTON 1



```
  F=X^2-5

  D=2*X

      ROOT               FUNCTION
  APPROXIMATION            VALUE
  -------------            -----

        2                    -1
      2.25              .0625000009
    2.23611111        1.92906708E-04
    2.23606798               0
    2.23606798               0

WOULD YOU LIKE ANOTHER
APPROXIMATION (YES OR NO)?
```

The Study of Realistic Applications

A program of value for science and social science classes as well as mathematics, LEAST SQUARES finds an equation (polymonial, power, or exponential) that best approximates the set of data pairs entered by the student. Real experimental data can be entered, the function determined by the computer, and predictions made using that function. The program calculates a measurement of the closeness of fit and displays a graph. Problems involving topics such as vapor pressure of water at various temperatures, solubility of compounds and radioactive decay of isotopes can be explored using this method. Many examples can be found in sociology, economics, and engineering as well as in science.

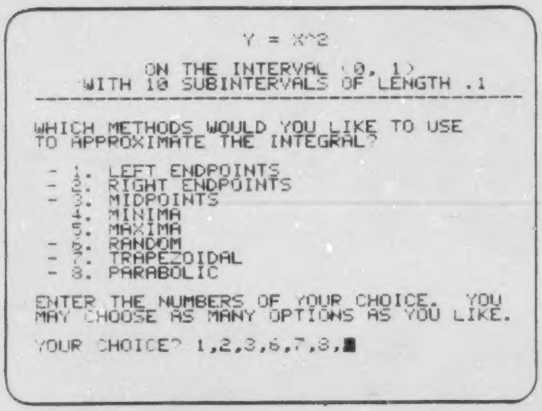
The MULTI-INTEGRALS program allows students to study problems relating to moments of inertia and volumes of solids without spending time on hand calculations of integrals. INTEGRATION permits exploration of situations such as the pollution levels in a lake and the cost effectiveness of new machinery in a manufacturing plant.

Experimentation with Various Approaches to a Problem

INTEGRATION provides eight different methods of approximating a definite integral, allowing students to choose any combination of methods and evaluate the results. Also, students can experiment with varying the number of subintervals used in the process.

The program MATRIX includes 15 options for operating on matrices. Students can solve systems of equations by various methods, for example by augmenting the matrix or by using both a coefficient matrix and a constant matrix.

INTEGRATION



```
              Y = X^2

      ON THE INTERVAL (0, 1)
   WITH 10 SUBINTERVALS OF LENGTH .1
---------------------------------------
WHICH METHODS WOULD YOU LIKE TO USE
TO APPROXIMATE THE INTEGRAL?

 - 1. LEFT ENDPOINTS
 - 2. RIGHT ENDPOINTS
 - 3. MIDPOINTS
   4. MINIMA
   5. MAXIMA
 - 6. RANDOM
 - 7. TRAPEZOIDAL
 - 8. PARABOLIC

ENTER THE NUMBERS OF YOUR CHOICE.  YOU
MAY CHOOSE AS MANY OPTIONS AS YOU LIKE.

YOUR CHOICE? 1,2,3,6,7,8,█
```

Manipulation of Variables and Observation of Results.

In many of these programs the student has the opportunity to change parameters and observe the resulting changes. For example, the graphing program allows students to change axes or view different areas of the graph, the integration programs include the option of changing the size of the subintervals, and LEAST SQUARES allows the choice of several different types of functions to fit the given data.

Conclusion

Courseware is available that enables students to broaden and deepen their understanding of many advanced mathematics topics. Instructors can introduce students to important numerical techniques without requiring students to have programming skills or to spend time writing programs. With these computational tools, students can focus on analyzing complex problem situations, formulating functions and algorithms to solve the problem, and interpreting and evaluating the results of the computations. Problem-solving courseware can, therefore, make advanced mathematics classes more relevant to students and more appropriate to their needs in a computer-oriented society.

THE ONTARIO PERSPECTIVE

Neal T. Emery
Chairman - Provincial Advisory
Committee on Computers in Education
Ministry of Education, Ontario
Mowat Block Queen's Park
Toronto Ontario
Canada M7A 1L2

ABSTRACT

In response to the impact of the new information technologies, the Province of Ontario, in Canada, has embarked on a number of computer-related initiatives. These include a multi-staged plan to develop exemplary lessonware and courseware through a special fund provided by the Board of Industrial Leadership and Development (BILD), criteria for the design and manufacture of a family of educational microcomputers, a mechanism for assisting school boards in the purchase of approved microcomputers, suggestions for co-operative planning, production of a computer studies guideline, and other activities. This paper will outline these initiatives and indicate other areas of exploration.

BACKGROUND

The Province of Ontario lies north of the states of New York, Pennsylvania, Ohio, Michigan, Illinois, Wisconsin, and Minnesota, and covers 412,582 square miles. 187 school boards operate 3,877 elementary schools and 637 secondary schools to provide education for about 1,750,563 students.

A Computers in Education Project was established early in 1981 by seconding personnel from various branches of the Ministry of Education to serve on the project. At the same time, a Provincial Advisory Committee on Computers in Education drawn from the Ministry and the educational community throughout the province was formed. This organization, including both the Advisory Committee and the Project Team, was intended to advise the Minister of Education on matters related to computer-based learning materials so that a paedagogical standard could be maintained throughout the schools of the province.

To help maintain this standard, several activities have been undertaken.

FUNCTIONAL REQUIREMENTS FOR MICROCOMPUTERS FOR EDUCATIONAL USE IN ONTARIO SCHOOLS

As of January 1, 1984, school board purchases of microcomputers and micro-computer systems specifically designated by the Ministry of Education as meeting these requirements, will be eligible for a rate of grant which on the average across the province, is approximately 75%.

The work leading to this statement began in the spring of 1981 when the Ministry of Education in cooperation with the Canadian Advanced Technology Association, collaborated to mount a preliminary study of the ideal characteristics of a microcomputer for educational use. Subsequently, the Ministry has been developing a formal statement as to the functional requirements which microcomputers for educational use must meet in order to be recognized as extraordinary expenditure.

To assist in developing this statement, the Ministry funded an additional consultant's study to recommend the priorities which should be assigned to the features identified in the earlier study. The Ministry has also solicited input, both generally and on specific issues, from individuals and groups with particular expertise. In consequence of this work, the Ministry has now completed a first statement of functional requirements. Because of rapid technological advance in the field of microelectronics, these requirements cannot be frozen but will be periodically revised and updated to reflect new capabilities as they become feasible. This first statement is, therefore, referred to as "Stage I". It will be replaced by "Stage II", and subsequently by Stages III and IV etc. as circumstances warrant.

Beginning in September 1983, the Ministry is prepared to accept for evaluation, computer systems whose manufacturers believe they meet the Stage I Functional Requirements. Such evaluation will involve internal assessment by the Ministry, pilot testing in classrooms, and technical bench-marking by an external testing organization designated by the Ministry. Systems accepted under Stage I criteria will be eligible for recognition as extraordinary expenditure so long as the Stage I requirements remain in force. This eligibility will be lost if the manufacturer is not prepared to upgrade Stage I systems to meet Stage II requirements when they come into effect. Stage II and later requirements will be pre-announced before the beginning of their period of validity, to ensure reasonable time for manufacturers to upgrade their systems.

Because both the Province and school boards will have an increasing substantial capital investment in educational softwares, the Ministry has felt it necessary to include in its requirements, specifications to ensure the maximum degree of "portability" or transferability of softwares from generation to generation of equipment. This will be achieved by carrying over from stage to stage certain provisions which will standardize the features of the user's interface and the programming languages.

PROVINCIAL ASSISTANCE TO SCHOOL BOARDS IN ACQUIRING SECOND GENERATION MICROCOMPUTERS: The Ministry of Education has entered into a contract to purchase from the Canadian Educational Microcomputer Corporation (CEMCorp) microcomputer systems which meet the Ministry's functional requirements to a total value of ten million dollars, for distribution to Ontario school boards. The current generation of microcomputers in Ontario schools, while fully adequate for the tasks to which it is now applied, will be displaced, both in education and business practice, by the newer generation of more powerful microcomputers based on 16 and 32 bit architectures.

The Ministry of Education wishes to ensure that microcomputers representative of this new generation and fully reflective of its functional requirements become available for use in Ontario schools. Two million dollars will be for prototype systems, deliverable in the summer and fall of 1983, and eight million dollars will be for fully developed systems deliverable in late 1983 and throughout 1984, together with the necessary systems software. The prototype systems will be pilot tested in Ontario classrooms and will become the property gratis, of school boards providing testing facilities.

The fully developed, or "commercial" systems will be made available to school boards at a cost which is 25% of the manufacturers commercial price.

The computer systems being developed by CEMCorp will be fully consistent with the Ministry's functional requirements. The systems covered by the Ministry's purchase agreement will be based on the following elements: i) A "Standard Student Microcomputer" based on a 16 bit electronic architecture with 128K of random access memory (expandable to 256K); ii) An "Advanced Student Microcomputer" based on 32 bit electronic architecture with 256K or random access memory, expandable to 1024K (available in the fall of 1984); iii) A "Fileserver" acting as a central storage facility for software (languages and sophisticated user interface) student personal storage and classroom management information, which contains a floppy disc drive, a hard disc drive, 256K of random access memory, networking electronics and ports for connection to printers, other peripherals and external networks; iv) A local area network including power and network cabling modules linking up to 32 standard and/or advanced microcomputers with the Fileserver; v) Video display terminals (12" monochrome, 15" resolution monochrome and 13" colour).

EDUCATIONAL SOFTWARE FOR ONTARIO SCHOOLS

It is essential that Ontario schools take full advantage of the contribution to learning that quality educational software can provide. This cannot happen unless lessonware and courseware which is expertly developed, in both the paedagogical and technical sense, and directly relevant to the learning objectives in Ontario curriculum guidelines, becomes widely available. For this reason, the Ministry of Education and the Ministry of Industry and Trade have collaborated, in cooperation with BILD, to institute a series of measures to assist the growth of an Ontario educational software industry.

1. An advisory body of representatives of the educational client groups and the educational software industry will be established to advise on the priorities to be observed in developing materials and the further development of the supporting program and policies of the two Ministries;
2. A process to identify and prioritize educational software needs will be established;
3. A program of developmental grants to educational software producers will be instituted;
4. The 'Crown in right' of Ontario will purchase licenses from educational software producers which will allow free and unrestricted use of materials in Ontario educational institutions but which will leave rights in other Ontario markets and in other jurisdictions, in the hands of the producers;
5. Provision of technical and paedagogical advisory services for software producers and to provide access to formative developmental evaluation;
6. Provisions will be made internally to identify potential producers, to develop evaluative criteria for assessing developmental proposals, purchasing licenses and accepting materials for cataloguing and distribution, to fund ongoing research, to integrate software needs with Ministry program development plans, and to affect general cooperation; and
7. The Ontario Educational Communications Authority will assume responsibility for establishing a cataloguing and dissemination service for Canadian educational softwares which meet the 2 Ministries' evaluative criteria.

To initiate these programs, BILD funds have been provided totalling $1.3 million in 1983 and $5.46 million in 1984, growing to $10 million by 1986.

These measures are designed to provide a supportive infrastructure within which a private Ontario industry can develop to supply in large measure the educational software needs of Ontario schools.

THE EXEMPLARY LEARNING MATERIALS PROJECT

The general program to promote and sustain software development outlined above had as a preliminary step the Exemplary Learning Materials Project which was established in order to identify individuals and companies interested in and capable of developing quality educational software directly related to the learning materials in Ministry of Education curriculum guidelines, as the nucleus for an educational software industry in Ontario, and in order to produce a range of materials exemplary of current best practice. In response to a Call for Proposals released in March 1982, approximately 510 project proposals were received. Teachers and computer experts were used to evaluate these proposals against three major criteria: excellence of instructional design; relevance to objectives in Ministry curriculum guidelines; and effective use of the computer. Fifty-seven specific lessonware projects were selected for funding. The development of each project is taking place in three stages: design, prototype and production, with an extensive evaluation both technical and paedagogical, at each stage. Successful completion of one phase is required for continuation to the next. It is expected that the majority of these projects will successfully be completed and that the resulting body of software will illustrate the current state of the art as envisaged by Ontario education and software producers interested in this field. This material will enhance the learning experiences of students in a variety of subject areas and, in a number of cases, will assist students in activities involving creative thinking. Many of these materials will be available in both French and English.

It is expected that the first materials will be available this spring, with the majority reaching completion in the fall.

COMPUTER STUDIES CURRICULUM

A new curriculum guideline for Computer Studies, setting out policy for the development of courses related to computers in grades 10,11 and 12, is in the final stages of production. Copies of the documents will be distributed to all schools by June 1983 to provide a resource for curriculum development teams and educators who wish to plan changes in current courses and design pilot courses to begin this fall.

Beginning September 1984, all Computer Studies courses must be based on this guideline which supercedes Curriculum RP33: Data Processing, 1966; Elements of Computer Technology, 1970; Computer Science Senior Division, 1970; and Informatics, Intermediate and Senior Division, 1972. This new guideline has been under development over the last two years and has been validated by a wide range of educators across the province, computer specialists and members of the Canadian "high tech" industry. It will provide for a wide variety of courses in computer technology, programming, information processing, and systems analysis and design to meet the varying needs and interests of students. Schools will be able to create alternate educational pathways for students who may wish to focus on job skills, life skills or academic endeavours. Related guidelines are planned (e.g.Ontario Academic Courses in: Computer Studies, Computer Technology) which will permit the development of complementary courses to enhance these pathways. The attached diagram (FIG. 1) provides an overview of the nature of programs which can be created by schools.

PLANNING FOR THE EFFECTIVE UTILIZATION OF COMPUTERS IN EDUCATION

Effective joint planning at the provincial and local levels will be required to ensure that technological resources are used to their full potential in integrated approaches which enhance classroom learning.

A. Planning at the Local Level

A Provincial Review on Instructional Use of Computers indicated that some boards have carried out extensive planning whereas others are just now addressing the various issues involved. Information collected indicates that school boards in developing plans should address the issues listed under the following broad headings: 1. The Educational Objectives for all Computer Initiatives; 2. Curriculum; 3. Computer Equipment: 4. Administrative Support Services for Board, School and Classroom; 5. Implementation Strategies & In-Service Plans including Outreach to Parents; 6. Resource Requirements.

B. Planning at the Provincial Level

"A Planning Report on Educational Computing in the '80's", developed as part of the work of the Provincial Advisory Committee on Computers in Education, outlines suggested initiatives for a number of planning strands.

Literacy - A draft statement will be prepared this year which deals with computer literacy for students, teachers, administrators and the public and the means which should be taken to achieve it.

Curriculum - Microtechnology will be integrated with curriculum as guidelines are revised and developed. Schools General, Ontario Schools Intermediate and Senior, and The Formative Years Renewal will include direction with respect to the New Information Technology.

Training - Pre-Service and In-Service initiatives to ensure computer literacy of teachers are underway at the faculties of education. A fact sheet will be prepared for the 1983-84 school year. The Ministry will continue its cooperation with OECA to ensure the wide availability of the extremely successful "Computer Academy".

Lessonware/Courseware - Key curriculum areas for the development of lessonware/courseware and instructional databases will be identified.

Cataloguing and Indexing - Criteria for the evaluation of computer-based learning materials will be developed along with plans for an Ontario database system. Discussions are being held with other provinces and the Council of Ministers of Education, Canada to tie in with a Canada-wide system.

Courseware Dissemination - A plan for the dissemination of software (lessonware/courseware) is being developed in conjunction with the Ontario Educational Communications Authority (OECA).

Research - Two studies have been funded by the Ministry: one by the Board of Education for the City of Toronto to look at technology use and teacher practice to identify some possible themes and/or relationships; the other at Queen's University to study the creative use of microcomputers.

Hardware - A strategy to revise and upgrade the Ministry's statement on "Functional Requirements for Microcomputers for Educational Use in Ontario Schools", in the light of developing instructional needs and technological developments over the next number of years, is being developed.

Funding - Specific funding arrangements to support hardware and software activities are underway.

C. Other Related Activities

A number of additional computer-related activities are underway throughout the province and should be considered, as plans to incorporate the new information technology into schools are developed.

Evaluation Guide - The Ministry of Education has funded the development of a guide for the evaluation of computer-based lessonware which will provide administrators with an outline of the procedures to be undertaken when evaluating computer-based learning materials at the board and teacher level.

Human Resources - In addition to a Ministry network, established through the regional offices and information networks such as that formed by the Educational Computing Organization of Ontario, various subject associations are being encouraged to form their own human resource networks.

Pilot Projects - Five school jurisdictions throughout the province are involved in pilot projects whose major purposes are: The identification of subject areas where computer applications have enhanced the existing educational program, the identification of the effects of the creative use of computers in the classroom on teachers and pupils; and the refinement of evaluation procedures for lessonware/courseware.

Alternate Delivery System - The joint venture involving TVOntario, Correspondence Education Branch, the Lake Superior Board of Education and the Computers in Education Project to establish effective ways to electronically deliver courses to isolated students is continuing. The courses being provided this year are Gr. 13 Physics and Basic Electronics.

Telidon - The Ministry has been working with TVOntario in the creation and delivery of an Educational Telidon database. TVOntario currently has over 21,000 Telidon pages including much of the Student Guidance Information System on their own database. Beginning in April 1983, a province-wide network of approximately 200 terminals will be installed in schools, Ontario Youth Centres and public libraries to evaluate the effectiveness of videotex systems in delivering youth-oriented and other types of educational information.

Many excellent initiatives have already been taken towards the effective introduction of computers in the the classroom. The programs outlined in this paper provide a sound but flexible base for future planning at both the provincial and local level. The new information technology can make dramatic improvements in the learning experience of students. Careful and cooperative planning can help ensure that we maximize these improvements and make them widely and equitably available to the students in elementary and secondary schools.

COMPUTER STUDIES COURSE ORGANIZATION

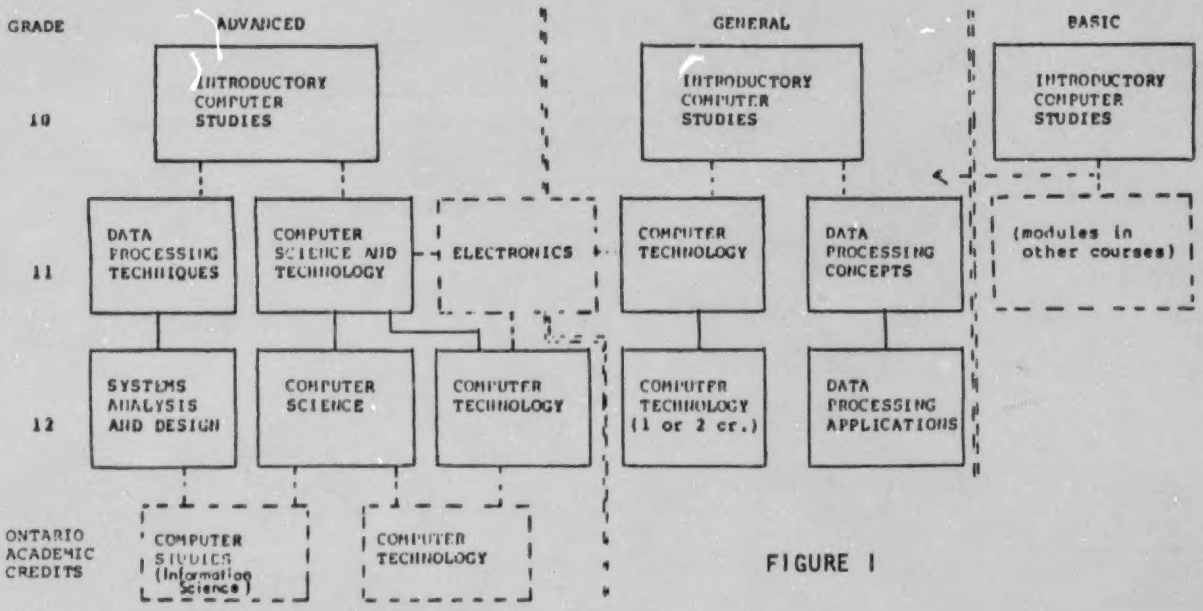

GRADE
ADVANCED
GENERAL
BASIC
10
INTRODUCTORY COMPUTER STUDIES
INTRODUCTORY COMPUTER STUDIES
INTRODUCTORY COMPUTER STUDIES
11
DATA PROCESSING TECHNIQUES
COMPUTER SCIENCE AND TECHNOLOGY
ELECTRONICS
COMPUTER TECHNOLOGY
DATA PROCESSING CONCEPTS
(modules in other courses)
12
SYSTEMS ANALYSIS AND DESIGN
COMPUTER SCIENCE
COMPUTER TECHNOLOGY
COMPUTER TECHNOLOGY (1 or 2 cr.)
DATA PROCESSING APPLICATIONS
ONTARIO ACADEMIC CREDITS
COMPUTER STUDIES (Information Science)
COMPUTER TECHNOLOGY

FIGURE I

Notes:

Each block represents authorization for a one-credit course except where noted otherwise. Courses in dotted boxes may be developed from other guidelines.
Dotted flow lines indicate recommended sequence.
Solid flow lines indicate recommended prerequisites.

LOGO: A COMPUTER LEARNING ENVIRONMENT

John J. Ferret, Jr.
Instructional Coordinator
Minnesota Educational Computing Consortium
2520 Broadway Drive
St. Paul, Minnesota 55113

ABSTRACT

New methods are being developed for young children to learn about and control computers using the Logo language. This paper, through three classroom activities, attempts to show this unique learning environment and its possible uses to develop a child's awareness of his individual thinking abilities.

The purpose of this paper is not to teach the fundamentals of the Logo language nor to be a directive for where educators should place Logo in the curriculum. What this paper really stresses is that Logo is a unique and new means for children to control the use of the computer and allow them to explore the fascinating learning environment that Logo provides.

Certainly, Logo is a computing language. However, it is unlike other languages such as Pascal or BASIC in that it was not created for the purpose of data processing. Rather, Seymour Papert and his associates at M.I.T. created Logo for the purpose of allowing children to communicate with computers more easily and in a way that will change how they learn other ideas. In today's classrooms the computer is mostly used to teach the child subject matter in a pre-determined learning sequence. However, Seymour Papert, in his book, "Mindstorms: Children, Computers and Powerful Ideas," states; "In the Logo environment the relationship is reversed: the child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think."[1] The key word here is exploration. Logo and the activities presented in this paper are a means for children to explore themselves and how they think.

To demonstrate how Logo does this, three activities will be presented. These activities were taken from the book "Apple Logo in the Classroom," developed by the Minnesota Educational Computing Consortium. The point to keep in mind is the degree of freedom allowed in the activity for the child to reach the desired outcome. There is not a set right or wrong answer in the activity. Rather, there is an environment of free intellectual expression for the child to attain the goal. Cynthia Solomon in an article that appeared in Byte magazine in August, 1982, on Logo, confirmed this when she stated: "This gives children the opportunity to express themselves and explore their own intellectual styles."[2]

In each of the activities students must be given sufficient time to explore and experiment. Also, group sessions should be a natural part of Logo activities to allow students to share with each other discoveries, analyze techniques, and develop concepts. The teacher's role in this process can be summarized as being "... a source of suggestions, reinforcer and questioner rather than a provider of answers. A Logo teacher must hold back the urge to say, 'This is how you ...' or 'The way to fix it is ...'."[3] If a teacher did provide the answers it would completely negate natural intellectual exploration for the child.

---

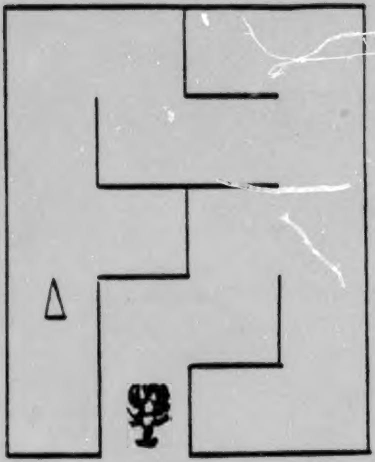
[1] Seymour Papert, Mindstorms: Children, Computers and Powerful Ideas, (New York, NY: Basic Books, Inc., 1980), P.19

[2] Cynthia Solomon, "Introducing Logo to Children," Byte, Vol. 7, No. 8 (August, 1982), P. 202.

[3] Apple LOGO in the Classroom, (St. Paul, MN: Minnesota Educational Computing Consortium, 1983), P. 4

Activity 1 - Mazes and the Turtle.

The objective of this activity is to move the "turtle," the triangular shape manipulated by Logo commands, through the maze pictured below to catch the crayfish.



A transparency of the maze can be made by the teacher so that it can be taped to the screen.

It is assumed that the students would have been involved with other activities prior to this one so they become familiar with some basic Logo commands. These prior activities would also have given the child the realization that the Logo commands, such as FORWARD, SHOWTURTLE, and BACK, can be abbreviated FD, ST, and BK respectively. Moreover, a child's prior work should have led them to understand the effect of RIGHT 90 or LEFT 90 (abbreviated as RT 90 and LT 90). Such introductory activities might include letting a child be a human turtle with the class suggesting commands to move the "human turtle" around the room.

With these considerations in mind, the students, either individually or in pairs at the computer, would move the turtle through the maze, noting on paper what they told the "turtle." Once this is completed, the students could meet as a group to discuss their results and discoveries, and develop ways to make their list of commands shorter. The following list can be considered an example of a child's first trip through the maze (at least around the first corner):

```
ST
FD 1
FD 10
FD 10
RT 90
FD 5
FD 2
RT 90
FD 10
```

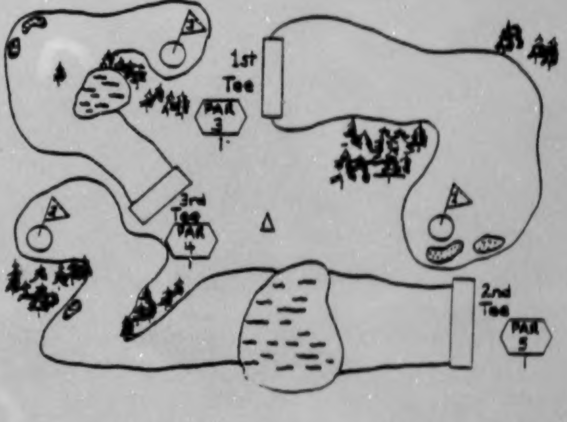
After the computer work and thorough group discussion these children may then see that they can shorten their work to:

```
ST
FD 21
RT 90
FD 7
RT 90
FD 10
```

The key, however, is that each child's independent exploration is neither right nor wrong. By sharing and questioning in a large group setting, the child can then return to the computer and try other ideas to shorten the command list. So, we see an activity, simple in nature, that allows young children to enter the Logo world, explore, discover and reach a goal in their own style with the use of the computer.

Activity 2 - Turtle Golf.

This activity would again use a transparency, this time showing a golf course:



The student(s) play a round of golf by manipulating the Logo Turtle from the tee until they "touch" the hole. One can find a full set of rules for the course and this activity in the book, "Apple Logo in the Classroom," referenced previously. However, the class as a group could decide upon its own rules and, possibly, even upon the course layout. Once again certain assumptions must be made, such as that the child has completed activities to discover the further uses of RIGHT and LEFT with angles from 0 to 360 degrees.

Once again the child would individually, or with one or two others, attempt to manipulate the turtle through the course, recording the command list used and keeping score of the game. It is again suggested that as the children reach the desired goal they would be, through their exploration, discovering shortcuts and refining their thinking for a future round. Once completed, the class could meet as a group, share discoveries, and learn from the other "Logo golfers."

A possible command list (depending upon screen size) to play the first hole might be:

```
FD 20
RT 90
FD 2
FD 4 (at the 1st tee)
FD 40 (hit)
FD 5 (hit)
RT 90
FD 30 (hit)
FD 10 (hit)
RT 20
FD 2 (final hit)
```

It is suggested that a child may begin to see at this time, after subsequent exploration or during group discussion, that other angles besides 90 should be used to keep the "score" under par. Clearly, further work by the child will lead him to independently discover the best moves to make and to develop techniques for the estimation of angles.

As in the first activity, the goals of the child controlling the computer and exploring independently the means to the end are clearly met. Also, it is clear that the availability of sufficient computer time for each child to successfully reach the goal is critical to using Logo. Children must have a chance to fully explore according to their own abilities.

Activity 3 - Colorful Building Blocks.

This final activity is quite different from the other two in that it is assumed that the student, through other activities in the Logo environment, would actually be able to "teach" the Turtle new words; in other words, to program using Logo. For example, the following Logo commands would draw a square:

```
ST
FD 40
RT 90
FD 40
RT 90
FD 40
RT 90
FD 40
```

However, with proper guided exploratory learning by the teacher, the child could be easily trained to write the following Logo program (actually called a procedure).

```
TO SQUARE : SIDE
REPEAT 4 [FD :SIDE RT 90]
END
```

The effect of this in the Logo world is that with one command the child could tell the turtle to draw a square of any size. The command SQUARE 40 would make the turtle draw a square 40 "turtle steps" on a side. The children would have also learned that they can save their program for future use in other programs on a diskette by typing:

```
SAVE "SQUARE
```

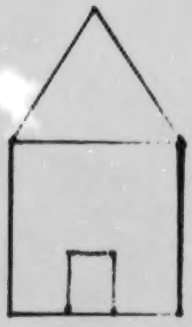
With these considerations in mind and assuming the child is capable of using the Logo editor, suppose a child has "taught" the turtle several new words (defined as a procedure) and saved them on a diskette.

These procedures might be;

```
TO SQUARE : SIZE
REPEAT 4 [FD :SIZE RT 90] draws a square
END                             of any size

TO RECTANGLE : LONG : SHORT
REPEAT 2 [FD :LONG RT 90 FD :SHORT
  RT 90]       draws a rectangle of any size
END

TO TRIANGLE  :X
RT 30
REPEAT 2 [FD :X RT 120]  draws an equil-
FD :X                  ateral triangle of
RT 90                            any size
END
```

The objective of the activity would be to have the turtle draw the student's dream house using color commands and the procedures they have already developed. The teacher should not necessarily specify the design, as this would put a restriction on the student's ideas and exploration.

The students would then enter the Logo environment to discover what they need to do using the building blocks they already have developed. A first creation might be a house as shown below:



using the procedure as listed;

```
TO HOUSE
CS
SETBG 1
SETPC 4
SQUARE 40
FD 40
TRIANGLE 40
BK 40 RT 90 FD 10 LT 90
RECTANGLE 30 20
END
```

The first creations will be quite varied depending upon the child's own individual expression. Using the building blocks already created, the child can delve into new discovery as they explore the Logo world and their own mind.

With the future of our own society so heavily dependent upon computer use, our schools must provide the means for children to learn about computers: how they work and, more importantly, what they can do. The Logo language, or rather the Logo learning environment, provides the child a unique way to attain this literacy and capability in addition to helping the child develop thinking skills. The Logo language should not be taken as a panacea for all educational problems. It is a powerful means to help our children learn about computers, about mathematics, about thinking, and about themselves.

References

Seymour Papert, Mindstorms: Children, Computers and Powerful Ideas, (New York, NY: Basic Books, Inc., 1980).

Cynthia Solomon, "Introducing Logo to Children," Byte, Vol. 7, No. 8, (August 1982), pp. 196-208.

Apple Logo in the Classroom, (St. Paul, MN: Minnesota Educational Computing Consortium, 1983).

BACH TO BASICS

Wm. P. Goddard
The University of British Columbia
Faculty of Education - Field Development Office
2125 Main Mall
Vancouver, B.C. Canada
V6T 1W5

ABSTRACT

Since 1965 a smorgasbord of software and strategy has been created in 'computers for the whole school'. The author has surveyed teacher choices through courses and workshops. Teachers have creative ideas for field developed programs using languages and utilities available to both themselves and student programmers. Teachers enter a 40 hour course with lesson plans and leave with programs developed for and by themselves. The best of BASIC, LOGO, PASCAL, and PILOT worlds are explored through applications projects, not lectures. Applications range from school administration to medicine. (Demonstration and handouts.)

INTRODUCTION

What do teachers really want from the computer? How may their opinions be obtained? What factors influence their opinions? A recent study shows that teachers favour a wide variety of materials ranging from the highly creative and open-ended to the more pedantic and highly structured(1). They are not convinced that any of the panaceas touted by over-zealous proponents of a single approach provides solutions for the complex problems in the infant field of computers in education. They have imaginative ideas for field developed courseware which, on the whole, they find more satisfactory than commercially developed courseware. They do, however, make excellent use of commercially written languages and utilities if they have training in these. The means of obtaining teachers' views outlined in the study show that questionnaires need not elicit many details regarding subjects as long as the target population for computer use is well specified and as long as all respondents have some computer experiences in common. It is this common experience in computing which is the chief factor in both the kind and quality of their opinions and suggestions. The argument has raged as to whether teachers can write instructional materials. It is the author's opinion that if this question is broken into two parts some progress can be made in finding answers. The two parts are: Can teachers design instruction after eighty hours of hands-on experience? Can teachers get students to program these designs to specifications if proper utilities are available? The author has attempted over the last three years to make 'yes' the answer to both questions through the vehicle of two courses. He has made many mistakes and learned several hard lessons, but feels that some progress has been made and that others may be interested in the current state of this program.

SETTING OUT THE SMORGASBORD
Each class has attended a locally convened orientation meeting at which all fees have been paid, registration forms and biography cards have been completed; machines, texts and materials assigned and scheduled and three documents distributed; Prerequisities and Requirements, Course Outline, and Evaluation Guide. Each class Group of three members must have access to a 48 K APPLE computer with peripherals, supplies, and texts in order to complete a pre-reading assignment (Applesoft Tutorial p.5-111) and in order to do assignments and to work on a 'project' (of immediate practical value to them) during the ten weeks of each course (one four-hour lecture-lab per week) and for one month after that. Each class is equipped with a demonstration machine, a duplicating service, printers, library and office.
Each four hour lecture-lab is comprised of two hours of talk (lecture, discussion, demonstration) and two hours of tutored hands-on work and student interviews. In addition each student is expected to devote at least four additional hours to hands-on independent work and preparatory studies each week.

THE FIRST COURSE (21 topics)
1. The Potential of the microcomputer for education. This is where BACH enters the picture. Demonstration begins with the ALF synthesizer playing Bach. A Chess game continues to show how 'intelligent' the computer is. A demonstration of MECC programs shows the potential for drill & practice, tutorial, and simulation modes of instruction. The class is shown how 50% of a teacher's time is taken up with activities which a computer might perform. Teacher anxiety is allayed as each member loads a disk of courseware and runs it. Members discover that, as teachers, they are capable of evaluating courseware without a detailed knowledge of computing. They are in control.
2.The Limitations and the Dangers through misuse of computers.
3.Looking Inside. Identifying the input, process/memory and output components. Modelling a program solution. Programming in immediate and deferred modes. The laws of precedence.
4.Printing Repeatedly- writing the first four programs to discover that the computer can print, calculate, repeat, decide, and store.
5.Calculating Repeatedly. Round-off, variables, predicting outcomes of PRINT with operators and strings.
6.Deciding. Flow charts & other devices for IF-THEN logic.
7.Storing, Initializing and Copying diskettes with DOS.
8.Looking inside the insides. Bits, Nibbles, Bytes. RAM, ROM, CPU. Decimal, Binary, Hex numbers, characters, and logic.
9.Tables, Scrolling, and Nested Loops. Logic and screen control.
10.Literals, strings, and response analysis. Instructor demonstrates a 'brute force' BASIC program called 'Provincial Capitals' while each group devises a piece of courseware on the same model. The instructor then provides a utility (like a mini-PILOT) called 'P.C.' to do the same task. Each group then converts its program in order to discover the joys and sorrows of using utility programs.
11.Learning by Experimenting. Each group attempts to have the computer explain itself by experimenting with a set of new commands. Experiences are shared.
12.Numeric and alphabetic arrays. The program 'MARKS' is

used to illustrate arrays and data files.

13.Task Analysis and Planning. Members discover the necessity for Top Down Analysis using 'Marks'.

14.Storing and Retrieving data using 'Marks'.

15.Format, Output, and Sorting using 'Marks'.

16.LOGO- or Are We Doing This All Wrong?

17.Word Processing, Printers, and MECC Master Utilities.

18.The Missing Links- for short memories. Three techniques when you run out of memory.

19.On Target & Curricular objectives. How to adapt courseware to suit local conditions.

20.The Computing Studies Class. Research findings and methods for producing students who can benefit themselves while implementing your designs.

21.A Project Ended- A Project Begun. Choosing and documenting your individual course project.

THE SECOND COURSE (21 topics)

1.Introduction to Principles of Instructional Design -Gagne & Briggs, 2nd edition, Holt.

2.Reinforcement and the events of instruction. Each class member applies Gagne principles to an evaluation of commercial courseware and then to the design of an individual course project.

3.Expanding flowcharts etc. into storeyboards. Each member presents project plans to class for help and criticism.

4.The modes of instruction in CAI. One group investigates PILOT and demonstrates and leads a class discussion.

5.CMI- another application of the Ed-micro. (HARTS, etc.) School administration and pupil tracking systems- class group.

6.CAL-What is this thing called (,) LOGO?-class group.

7.Historical Development-video series & book, 'Connections'.

8.Impact on Society- is this 'knowable? Are we ready for anything?

9.Peripherals- A class group investigates and reports, demonstrates and leads discussion of Visicalc, light pen, Heuristics Lab, etc.

10.Categories of Languages. Class groups investigate, demonstrate, and lead a discussion on PASCAL.

11.A few basic tricks-sharing of techniques and project work.

12.ABS & other functions. More tricks common in business computer programming.

13.EXEC, Editing (TOOLKIT) and a picture of memory- groups.

14.Getting the Picture & other utilities; Master, Toolkit, Tables, Organizer, etc.

15.Moving the Picture & Music.

16.Evaluating Courseware- ours and others. Is formal evaluation valuable?

17.Exemplars in the Humanities. The Management of a Data Base. 'Hypothesis Tester', 'Literature' programs.

18.Exemplars in the Sciences. Solving format and presentation problems in mathematics and the sciences.

19.Vocational Training and 'CHOICES'. There is a compromise between 'general' and 'vocational training'. Specific job training as, say, a Data Entry Clerk, may well fit the curriculum in that students who know they can cope in the job market may gain patience to continue general education. Students can solve real problems and begin to undertake original 'systems analyses'.

20.Video disks and networking. The computer as a general purpose audio-visual aid under logical control.

21.Advanced Word Processing and interviews to aid in the preparation of final projects.

CLEARING THE TABLE

These courses are demanding, but the rewards are equal to the effort. We are all pioneers in a new field. When one considers the changes wrought by the Book (which is only an extension of man's memory), consider what will be accomplished by the Computer (which is an extension of man's reason).

Students in the second course undertake a systems analysis and solution for a problem in a field outside their own as well as a major project. In this way it is hoped that they will join that cadre of resource persons which we need so badly in the field of Computers in Education.

THE DESSERT- projects

Description-Demo incuding:
Fun With Sets
Hypothesis Tester
Super Bee
Kelowna Studies
Man's Shelters
Grade Five Literature
Chemistry Cross-Over Method
Explorer Experience- discovery of North America
Bits and Pieces of Fractions
Physical Fitness Tests
Cognitive Tests for Tennis
Arrowmath
Diagnostic Computational Testing
An Introduction to Vectors
A User Modifiable Sentence Fault Testing Drill and Diagnostic Tool
ESL- A New Program
Farmersade
The Florist's Almanac
LOGO Intro for Special Needs
The Real World of Computers
Space Simulation-Projectiles
Hypothesis Tester-Third World
Weather & The British Isles
HARTS & MEDICINE

ACKNOWLEDGEMENTS:

Much of this material owes its existence to my colleagues at UBC: Drs. Seong-Soo Lee, Marvin Westrom and Peter Braun on the academic side; The Computer Application in Education Committee, for background; Douglas McWhannel, Barbara Brown and Ron Neufeld for logistics; Stephen White and Robert Bruce for technical aid.

Over three hundred teachers since December 1980 have suffered the growing pains as this material developed and took the author at his word by telling him what was wrong with it.

The writers whose work has most recently influenced this work, (though no fault is their own) have been Robert Gagne, Seymour Papert, George Gerhold, David Moursund & Alfred Bork.

The greatest inspiration for this work came from school students who shared our first faultering use of 'Computers for the Whole School' from 1965 to 1980. They proved that they were capable of the production of valuable work if they had teachers who would only let them.

****

(1) Goddard, Wm.P. & Pereira-Mendoza, L., "Three Perspectives for Computer Applications in Education." (In process).

ADMINISTRATIVE PLANNING AND IMPLEMENTATION FOR INSTRUCTIONAL COMPUTING

Eileen K. Gress
Fairport Central School District
140 Hulburt Road
Fairport, New York 14450

ABSTRACT

A five-stage development cycle is presented as a model for successful implementation of an instructional computing program. Emphasis is placed on practical administrative strategies which are based on research and experience. Events in each stage are related to organizational requirements; new role relationships, strategies and tasks; and staff development. Particular attention is paid to the need for planning resource allocation and staff development based on specific computing applications.

This is a critical time for instructional computing. Though computers enter the school place at an ever increasing rate, we have yet to see a broad impact on the quality or quantity of learning. Indeed, it is the educational plan that will largely determine the ultimate effect of computing on student learning. Without a properly conceived and executed plan, we commit the crime of GIGO - Garbage In, Garbage Out.

A five stage development cycle, including Readiness, Planning and Resource Allocation, Training, Implementation, and Evaluation, can be used as a model for instructional computing programs. The model is effective not only for innovations, but for operational programs. Since an omitted stage may presage implementation difficulties, the process should be completed even if it means reverting to an earlier stage.

Traditional development models primarily attend to the written product. The tendency to be narrow minded, however, must be overcome if computer technology is going to have a major effect on schooling. Implementors must be mindful of organizational changes other than curriculum. Fullan and Pomfret (1977) suggest four other dimensions that must be attended to: resource allocations including grouping, spatial arrangements, personnel to perform new roles, new materials, and allocation of time and space; new role relationships, strategies, and tasks; knowledge and understanding that the users have about the innovation's components; and value internalization. I shall address each of these concerns through the use of the development cycle outlined above.

READINESS STAGE

Program implementation cannot be achieved without creating an environment which augurs success. Two goals are essential for achieving readiness. Each public of the district should become aware of the components of instructional computing and should develop a positive belief system, including a desire to learn about the subject and the belief that the computer is an effective teaching tool.

Three variables must be considered: the facilitator, the various publics of a school district, and the activities used to achieve the goals of the stage. Internal personnel or local expertise can be used to provide awareness sessions. These publics should be addressed: faculty, administrators, school board, and community. Successful strategies include workshops, board presentations, and parent-teacher association workshops. Demonstrations to students may be the most effective strategy of all. Student interest creates a domino effect; they motivate parents, who have immediate access to the board's collective ear.

Each public should be made aware of

the forms of instructional computing and have experience in viewing and using software. Demonstrations should include samples of each type of computing application. I suggest that the following characterize the instructional uses of computers: the computer as an object of study, as an aid to instruction, as a management tool, as a machine to be programmed, and as a tool for solving problems.

Beyond this core content, awareness sessions should address the distinct needs of each audience. This enables the innovation to become personally meaningful (Miller & Wolf, 1978, p. 142). Faculty must be apprised of the program's potential effect on the teaching process, including the need to reorder instructional priorities and increase their roles as facilitators of individualized learning.

Administrators should be acquainted with the management aspects of program implementation and be advised of the need for administrative commitment to allocate funds and facilitate staff retraining. The support of the building administrator is critical and must be generated at the outset. Indeed, the principal's contribution lies in his/her providing the project with "legitimacy" by supporting it (Berman & McLaughlin, 1978, p. viii).

As a policy-making body, the school board must be informed of the need for a long-range commitment. The board must be willing to commit not only funds, but time to review implementation plans so that an equitable program is achieved.

PLANNING AND RESOURCE ALLOCATION

It is at this stage that a framework for long-range planning should be developed. Rand Corporation's study of federal programs supporting educational change (McLaughlin & Marsh, 1978) found that ambitious projects were either met with indifference, resistance, or short life if the planning was either grass roots or district office dominated. Thus a broad-based committee, including faculty and administrators, should be appointed by the superintendent to develop recommendations for program goals, budget and implementation timeline.

Instructional goals should be identified through a needs assessment which includes a survey of staff and community perceptions of the relative importance of computing and an examination of current wisdom about the forms of computing. The assessment should assist the committee in determining the extent to which each form of computing should be introduced in short- and long-range plans.

A patchwork curriculum must be avoided. In order to produce an articulated and coordinated curriculum, a philosophy statement must be generated so that specific program decisions can rest on a sound foundation. Philosophy should include program rationale, a statement regarding staff responsibilities and a definition of program organization. Decisions include:

* Is computing part of the core curriculum?
* Who is responsible for teaching computing in the non-departmental elementary school?
* Who is responsible for computer instruction in secondary schools?
* What organizational structure will best allow for district coordination, building coordination, program equity, and technical assistance?

Once such philosophy is achieved, curriculum development should be an outgrowth of program goals. The Fairport Central Schools employ a core curriculum (1981) which includes careers, components of a computer, applications, social implications, flowcharting, history, interaction with a computer, and introductory programming concepts. In addition, computing strands for talented students begin in grade four; these include programming skills, problem solving, and analysis of social issues. Electives in computer science begin in grade eight. In addition, computer applications are infused in all areas and a formal CAI remedial reading program was initiated this year.

Courseware and hardware acquisitions are dependent on curricula. The opposite occurs too often; indeed, many districts are responding to a technology pull rather than a needs push. Courseware should not only be evaluated for its effectiveness, but for its feasibility of use. Such issues as how many staff members will employ the software and how long it takes teachers to learn to operate the program effectively must be considered. Implementation issues are equally important when purchasing hardware. Though different instructional program goals may dictate different hardware configurations, issues such as teacher training and hardware maintenance may preclude hardware diversity.

The culminating aspect of the planning and resource allocation stage is, of course, the actual budget. Non-budget items must be considered as well; adequate time must be available for teacher training, preparation, instruction and practice.

TRAINING STAGE

With curricula defined, staff development objectives can be formulated. Objectives should address knowledge, values, teaching strategies, and new teacher roles. This is a sizable task, one which is too often ignored by budget-conscious administrators.

The Fairport Schools use a hands-on group problem solving approach as the basis for staff development. Three levels of fifteen-hour courses are offered. Building-based workshops are also provided; these address topics defined by teachers or administrators (see Figure 1).

This training stage ties together much of the planning completed during the earlier stages. Staff development, however, should continue into the fourth stage, when the program is implemented.

IMPLEMENTATION STAGE

The thrust of this stage is "...to provide an environment that will support the transition of inservice learnings from the workshop setting into the daily activities of classroom teaching..." (Wood et al., 1981, p. 82). New teaching strategies and roles are best rooted when on-the-job assistance is available.

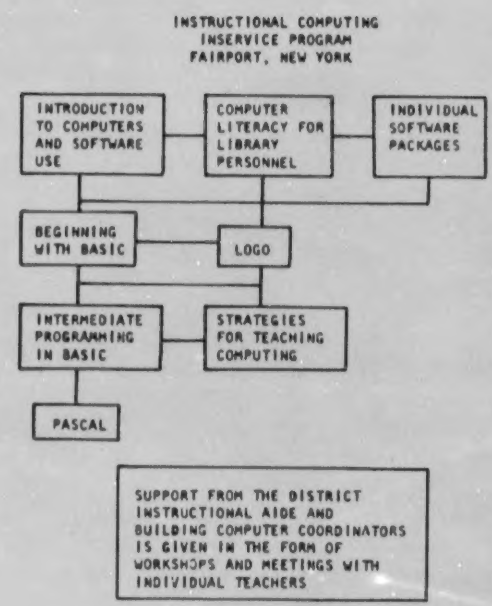

INSTRUCTIONAL COMPUTING
INSERVICE PROGRAM
FAIRPORT, NEW YORK
INTRODUCTION TO COMPUTERS AND SOFTWARE USE
COMPUTER LITERACY FOR LIBRARY PERSONNEL
INDIVIDUAL SOFTWARE PACKAGES
BEGINNING WITH BASIC
LOGO
INTERMEDIATE PROGRAMMING IN BASIC
STRATEGIES FOR TEACHING COMPUTING
PASCAL
SUPPORT FROM THE DISTRICT INSTRUCTIONAL AIDE AND BUILDING COMPUTER COORDINATORS IS GIVEN IN THE FORM OF WORKSHOPS AND MEETINGS WITH INDIVIDUAL TEACHERS

FIGURE 1

The Fairport Schools provide such an environment in two ways. An ongoing committee is used as a communicational link among buildings. These teachers are available to assist their colleagues in implementing the curriculum. Second, on-site assistance is provided by district staff on the basis of teachers' expressed needs.

EVALUATION STAGE

With a program in place, we educators often conveniently omit the evaluation stage. Yet it is primarily this stage which "establishes continuous monitoring to determine whether new behaviors are still being practiced and goals met" (Wood, et al., 1981, p. 86).

Though student achievement is a primary evaluation question, it does not become important until the level of implementation is determined. Investigating whether the program is being implemented with fidelity can turn up flaws in the process. Perhaps poor attitude is causing a low degree of implementation. Or lack of on-the-job assistance may prevent staff from adopting new roles. Staff, students, and parents should be queried to determine the amount of congruence between stated objectives and actual instruction. The implementation process can then be altered as needed.

I have reviewed five stages as if they were discrete processes. Indeed, they are not. Rather, they present a convenient way to describe and plan a process. In reality, the stages overlap and should be revisited. Yet they aid the educational planner by providing a set of considerations to help guide decision making. Instructional computing has characteristics that are unique to that innovation. We cannot enter that world blindly.

REFERENCES

Berman, P., & McLaughlin, M.W. Federal Programs Supporting Educational Change, Volume viii: Implementing and Sustaining Innovations. Santa Monica, California; Rand Corp., 1978. (ERIC Document Reproduction Service No. ED 159-289)

Fairport Central School District. K-12 Computer Education Curriculum, Fairport, New York: Fairport Central Schools, 1981.

Fullan, M., & Pomfret, A. Research on Curriculum and Instruction Implementation. Review of Educational Research, 1977, 47(2), 335-397.

McLaughlin, M. W., & Marsh, D. D. Staff Development and School Change. Teachers College Record, 1978, 80, 69-94.

Wood, F. H., Thompson, S. R., & Russell, F. Sr. Designing Effective Staff Development Programs. In B. Dillon-Peterson (Ed.), Staff Development/Organization Development. Alexandria, Virginia: Association for Supervision and Curriculum Development, 1981.

WORD PROCESSING FOR THE JUNIOR HIGH STUDENT WITH WEAK WRITING SKILLS

David W. Griffith
Assiniboine South School Division
3401 Roblin Boulevard
Winnipeg, Manitoba
R3R 0C6

This paper reports on a project that involved the use of a portion of the "SCRIPSIT" word processing package under the direction of a resource teacher and resource aide in cooperation with the classroom teacher to improve the writing skills of junior high students selected for their lack of ability to write their ideas in complete and correct sentences. Details concerning the necessary hardware, resource staff training in computer operation, procedures for insuring classroom teacher cooperation, the use of student tutors, and the success of the project to date as measured by actual student writing samples will be discussed.

Background

There exist a number of students, who at the junior high school level, are unable to express themselves in written form. Where their abilities are such that they are functioning successfully in many areas of their academic programs other than those areas requiring expression in written form, it appeared possible to utilize the microcomputer to approach their writing problem in an innovative manner and to use student excitement about computers to advantage in improving their writing skills.

The school resource teachers, in cooperation with language arts teachers, were asked to identify a number of students who were not expressing themselves clearly in written form. Standardized test results or survey test results were not used to screen those referred to the project for assistance. Students were referred to the project for a wide variety of reasons. Some students had obvious difficulties with the mechanics of handwriting or printing; others seemed unable to attend to the task of writing for a time sufficient to achieve success; others were satisfied to attend to all communication needs in verbal expression. However, the writing skills of all students involved in the project evidenced a considerable lag when compared with the writing skills of their peers.

Project Site

The school in which the project took place is located in a suburban area of Winnipeg, Manitoba. It is a moderately sized junior high school with a student population of approximately 600 and a professional teaching staff of 36.75.

Computers were first introduced into the school in the spring of 1982 when a "computer literacy" module of some twelve instructional hours was implemented with grade eight classes. Staff members who were to teach this module were provided with a seven session (of two hours per session) pre-service training course to prepare them to teach the material. During the 1982/83 school year, students in grades seven and eight were given this same module of instruction while grade nine students were given the second level of this "computer literacy" program. The school has the use of a network system of Radio Shack Model III microcomputers during the time when the course is offered at the school. Thus, all students and staff members involved in the delivery of the computer literacy modules are somewhat familiar with the operation of the Radio Shack Model III microcomputer.

"SCRIPSIT" Limited Applications Manual

During the planning stages of the project, it became obvious that only a limited proportion of the "SCRIPSIT" word processing software was desirable for use by the student clients who would be involved in the program. The use of "SCRIPSIT" should be as simple as possible if the attention of students was to be directed to the task of WRITTEN EXPRESSION and not directed to the OPERATION of the word processing package. As well, the professional staff involved in the project would be novice computer users. These characteristics demanded that the user friendliness of the software be extremely high. A simplified manual that covered only a selected number of the total features included in "SCRIPSIT" was written and used as a basis for the project.

As well, an early decision was made to have the student client supported by another person while using the computer. It was felt that this would be necessary to permit energies to be directed at the writing process instead of machine operation. Given these decisions, the "SCRIPSIT" LIMITED APPLICATIONS MANUAL was written.

The following features of the "SCRIPSIT" package were included:

* Load Scripsit into RAM
* Correcting Typing Mistakes
* Removing Unwanted Letters or Words
* Adding Letters or Words
* Setting up a Document Title
* Spacing Between Sections
* Exchanging Words and Paragraphs
* Saving Your Text on Disk
* Loading Your Text from Disk
* Printing Your Copy

The manual was written to support the adult who would act as "tutor" during the operation of the computer by the client. Much of the detail in the manual is necessary for the "tutor" but need not be known by the client. Experience in using the manual has resulted in several revisions for added clarity.

Pre-Service Training

The amount of training time required to train a novice computer user to be prepared to use the "SCRIPSIT" package with students was found to be approximately two hours of instruction where each trainee had a microcomputer for his own use. Following the training session, an additional four hours of actual use of the program by the trainee was needed to permit a comfortable level of familiarity with "SCRIPSIT" to develop. In addition, telephone "crisis" consultation was very helpful as trainees were experimenting in their first hours of use.

Initially, two resource teachers were trained in the use of "SCRIPSIT." The project school has as permanent computer equipment the following hardware:

Radio Shack Model III 48K with
Two Disk Drives

Radio Shack Line Printer VII

This equipment is located in the school's resource centre.

Project Stage One

In the first month of operation, the two resource teachers each took on one client and attempted to work with the client during at least two class periods per week. The initial sessions were spent in developing some degree of familiarity with the word processing commands: cursor moves, deleting a character, inserting a character, inserting a word or phrase, saving text, and loading text. Student interest in using the computer was very high, and with the resulting motivation, both students were quick to develop a functional level of skill in operating "SCRIPSIT."

During the subsequent sessions, the students involved in the project worked with their resource teacher in generating writing assignments that were printed at the end of the sessions and taken back to the classroom teacher for consideration as the student's assignment. In general, the writing produced by the students showed gradual improvement as the sessions progressed. One student was observed to have almost no errors in the material that he was writing on the screen after only two sessions. When this was discussed with him, he indicated that he "had many more errors in writing with a pen because his pen always blew up because he doesn't like writing; he doesn't like writing with a pencil which he can erase and pens are messy to go back to..." This student was making corrections as he typed in his writing.

The second student made less dramatic improvements in writing quality but showed several significant gains from an early point in his writing. When he began, he totally ignored the mechanics of writing in the area of punctuation and capitalization. This was discussed with him once, the problem being displayed on the screen and the resource teacher and student working cooperatively in making the necessary corrections. Subsequent writing on the screen contained almost no punctuation or capitalization errors. This topic had been covered frequently in his language arts class, but with no success. Another key consideration in assessing the progress here is to know that this student is almost impossible to maintain successfully in the classroom setting. His disruptive behavior is a constant concern in the classroom and contributes to his learning difficulties. When working on the computer, he attends to his task without excessive need for adult input for an entire class period!

At the conclusion of the first month of operation, an informal progress assessment of the project was undertaken to determine any necessary changes in implementation strategy. The following adjustments were made.

1. The amount of time dedicated to the project by the resource teachers was seen to be excessive with reference to all other aspects of their job. It was seen as necessary to train the two paid school resource aides in the

operation of "SCRIPSIT" in order to permit them to assume some of the one-to-one work with the client students. As well, the aides were working with these students inside the classrooms and the "SCRIPSIT" support would be a logical expansion of their involvement.

2. The degree to which the classroom language arts was being involved was not sufficient. It appeared that the students working on "SCRIPSIT" were not having their word processing work integrated into the normal course work. The effect of this was to put undue pressure on the resource staff to do all the pre-writing activities with the students and to make it difficult to work on the "transfer" of the word processing writing gains back into the classroom.

Project Stage Two

The changes in strategy noted above were incorporated in the work that continued. Two resource aides were trained. The classroom teacher was approached and the need for greater cooperation was explored. In addition, a number of student tutors were trained to support a related project involving the use of Computer Assisted Learning programs. It is intended to begin the necessary training in "SCRIPSIT" with these student tutors when more students are to work with word processing.

Interim Conclusions

The project is continuing at the time of writing of this paper. A more complete discussion of the project will be possible in the oral presentation to be made at the AEDS Conference in May. At this point, however, some interim conclusions and recommendations are possible.

1. The delivery model must be changed to more effectively integrate the "SCRIPSIT" support into the classroom activities. The key person to train in the use of "SCRIPSIT" is the classroom language arts teacher. Further, the computer must come to the classroom instead of having the computer in a location removed from the classroom.

2. A major hindrance to student use of the computer is a lack of sufficient familiarity with the keyboard. This may be addressed by including more emphasis on keyboarding skills at the seventh grade level in the "computer literacy" modules delivered to all students.

3. The clients selected appear to have shown some improvement in writing skills during the time of this project. However, many other factors contributed to the growth in skill. In order to capitalize on the powerful motivation provided by the computer, more or preferably all students should have access to the word processing experience. Here, also, it may be possible to include initial word processing training as a part of a "computer literacy" module delivered to all grade seven students.

4. Once the final project assessment is completed, and if the results are as positive as anticipated, it will be recommended to increase the level of hardware support in system junior high schools for the purpose of word processing as a component of the language arts program.

DIFFERENCES IN COMPUTER AWARENESS BETWEEN EDUCATION AND BUSINESS MAJORS

F. A. Griswold, M. Miller Griswold, G. Mobley
Eastern Montana College
Billings, MT 59101

ABSTRACT

Business majors were considerably more aware of computer applications and implications than education majors. These differences in awareness were attributed to greater experience and familiarity with computers. Emphasis on information access in education curricula, modeling the business curricula, may narrow the difference.

This study was based upon the premise that in the current Information Age, educators have the responsibility for ensuring that young citizens will have realistic attitudes toward and knowledge of the role of computers in our society. The major concern developed from evidence suggesting that educators lack computer awareness (e.g., Stevens, 1980). This lack of awareness may be related to negative attitudes toward computers (Clement, 1981; Huntington, 1981) and/or to limited exposure during academic training. With the possible exception of the recent implimentation of computer literacy requirements through education course offerings, education students have had very little opportunity to use computers or to conceptualize how computers might be used to further their teaching goals.

For those in the field of business, the opposite seems to be true. In fact, the field of business might be considered a leader in computer use and application. Business recognizes that technological progress contributes to productive growth (Lewis, 1982) and that effective decision-making is dependent upon timely and efficient access to appropriate information (Ungson et al. 1981). As a part of management theory, business students are trained in management information systems (MIS). Computerized MIS provides acquisition and dissemination of timely information, which serves as the basis for organizational planning and decision-making (Swanson, 1982).

Several studies suggested that personal characteristics, mathematics ability and psychological factors may be related to attitudes toward computers (e.g., Bozeman, 1978; Griswold, 1983) and to effective computer use (Cheney, 1980; Tillman, 1979). These factors may help explain the apparent discrepancy in computer awareness between the fields of business and education.

Given the different kind of training students receive in these majors, the somewhat negative attitudes held by educators, and the possible relationship with psychological measures and math abilitities, the following hypothesis was developed: Students majoring in business have a greater awareness of computers and their applications than students majoring in education. In addition to testing this hypothesis, the relationship between computer awareness and the following variables were studied: age, sex, locus of control, math experience and math ability.

Methods

Subjects

Participants in this study (N-240) were students attending the first week of class meetings in required courses. Eleven individuals were omitted because they were education majors taking business management courses. Among the 229 remaining participants, 62% were education majors (of these 44% were masters level students) and 38% were business majors. The mean age was 27.2 (SD=8.6) and females represented 62% of the total.

Instruments

Computer awareness was determined by a 20 item questionnaire addressing applications and social implications of computer technology (CAQ). The items fell into four categories. First were the computer's impact on the individual, e.g., "I am familiar through my previous experience", "Someday I will have a computer in my home". Second were attitudes toward computers, e.g., "Computers dehumanize society", "Are beyond the typical person's understanding". Third was a category of general computer capability, e.g., "Computers are a tool", "They can improve health care, law enforcement and education". The final category was educational applications, e.g., "Computers could teach math, reading, remedial instruction, take over parts of my courses". Each item was rated on a 5 point Likert scale ranging from strongly agree to strongly disagree. A composite score was created by summing the 20 ratings. The reliability of the CAQ was high among all 240 who were administered the instrument (Cronbach's alpha=.84).

Several measures of mathematics preparation were collected. Ranks for the number of math courses taken in high school and in college were used to measure math experience. A ten item basic mathematics skill test was administered to estimate math ability.

Since locus of control was shown to be a predictor of attitudes towards and awareness of computers (Coovert & Goldstein, 1980 and Griswold, 1983), it was also included in the testing schedule. The Rotter I-E scale (Rotter, 1966) was used to measure the degree to which a person perceives events to be unrelated to their behavior.

Procedures and Analysis

The frequency of responses to each of the items on the CAQ was computed for business and education majors. Since agreement with items having negative connotations was taken as an indication of lack of awareness, the direction of these ratings was reversed to be consistent with items having positive connotations. Multiple regression analysis was used to predict CAQ composite from all other variables.

Results

Business majors (BM) demonstrated that, on 17 of the 20 items from the CAQ, their awareness was greater than education majors (EM). The expected frequency of responses for BM on 9 of these items was significantly different from EM.

These differences reflect greater awareness by BM in three categories: 1) the computer's impact on the individual, 2) the affective impact, 3) general computer capabilities. Two striking differences were found related to items on the impact of computers on the individual. For example, 92% of BM indicated previous experience with computers, while only 38% did so among EM. Eighty percent of BM predicted having a computer at home, but only 48% of EM made a similar prediction.

EM reported levels of awareness similar to BM on specific educational applications of computers, e.g., computers could teach math and reading, do routine tasks, enhance remedial instruction, and make them a better teacher. Also EM were similarly aware that: computers may take their job; influence of computers cannot be escaped; and computers will replace low-skill jobs with jobs requiring specialized training.

To compare overall differences of awareness between majors, the mean CAQ composite scores were subjected to analysis of covariance. Since externality and math skills were significantly different by major, they were used as covariates. The results indicated that when controlling for the effects of externality and math ability, mean CAQ composite scores for BM (Adj. $\bar{X} = 17.75$) was significantly greater than EM (Adj. $\bar{X} = 10.91$).

When computer awareness was regressed on all variables, 28% of the variation was explained (i.e., multiple r = .53, p .0001). The variables age, major, and ranking of number of high school and college math courses were the significant predictors of CAQ composite (beta = .26, .35, .19, and .16, respectively). They accounted for more than 89% of the explained variance. The variables sex, externality, and math skills test did not make a signigicant contribution.

Discussion

The hypothesis that the awareness of computer applications among business majors was greater than among education majors was supported. On questions of personal impact, general capability, and affective perceptions of computers, business majors reported greater awareness than education majors. On items relating to educational applications, affect on jobs, and the pervasiveness of computer influence, business majors did not exceed the awareness of education majors. Overall, business majors demonstrated a much greater level of awareness of computer applications than education majors.

Choice of business as a major (which presumes preparation for computer application), age of students, and, to some extent, math preparation were important predictors of computer awareness. These variables all contributed to a greater probability of exposure to computers. Business curricula emphasize computer use in decision-making. With age, greater life experience presumes exposure to information about computers. And, math preparation historically has been a prerequisite for computer use in high school and college. Thus, exposure to computers or information related to computers, may enhance awareness of computer applications more than previously identified factors.

Several implications of this study may be drawn. Business, with a continued supply of computer-prepared graduates, will maintain a leadership role in the applications of computers. Education will continue to lag behind. To narrow the gap and directly influence educational applications, educators and prospective educators must become more experienced with computers. This could be achieved by requiring more math courses in which computers are used and the integration of computers and information systems procedures into education curricula. A required computer literacy course, like those for teacher certification, will not be adequate. The curriculum in education requires an integration of computers like any other information tool. As objective educators we should look closely at business curricula -- we may learn something.

References

Bozeman, W. C. Human factors considerations in the design of systems of computer managed instruction. *Association for Educational Data Systems Journal*, 1978, *11*, 89-96.

Cheney, P. Cognitive style and student programming ability: an investigation. *Association for Educational Data Systems Journal*, 1980, *13*, 285-291.

Clement, P. Affective considerations in computer-based education. *Educational Technology*, 1981, *21* (4), 28-32.

Coovert, M.D. and Goldstein, M. Locus of control as a predictor of users' attitude toward computers. *Psychological Reports*, 1980, *47*, 1167-1173.

Griswold, P. A. Some determinants of computer awareness among education majors. *Association for Educational Data Systems Journal*, 1983 (in press).

Huntington, J. F. The impact of changing computer resources on educational institutions and computer-based education training programs. Educational Technology, 1981, 21 (10), 55-59.

Lewis, J. D. Technology, enterprise, and American economic growth. Science, 1982, 215 (5), 1204-1211.

Rotter, J. B. Generalized expectancies for internal versus external control of reinforcement. Psychological Monographs, 1966, 80 (Whole No. 609).

Stevens, D. J. How educators perceive computers in the classroom. Association for Educational Data Systems Journal, 1980, 13, 221-232.

Swanson, E. B. Measuring user attitudes in MIS research: a review. Omega: The International Journal of Management Science, 1982, 10 (2), 157-165.

Tillman, M. R. An examination of the productive validity of several potential predictors of the work proficiency of computer programs. Computer Personnel, 1979 (Aug.), 3-6.

Ungson, G. R., Braunstein, D. N., & Hall, P. D. Managerial information processing: a research review. Administrative Science Quarterly, 1981, 26, 116-132.

"Effective Teacher-trainer Programs: Of What do they Consist? A Review of the Research."

Monica C. Hamolsky
Center for Children and Technology, Bank Street College of Education
610 West 112th Street
New York, NY 10025

Abstract: Recent research surveys show that teachers frequently find computer training through coursework inadequate. In an effort to construct criteria against which training programs may be evaluated, this paper will survey current research on systems delivering professional preparation. Critical conditions of adult learning will be examined, including those interaction patterns and instructional supports contributing to institutionalization of training.

Contextual Frame. In 1982, 145,000 microcomputers were registered as having been sold to schools, and it is projected that approximately 500,000 will have been purchased by 1985. At that time, sale of educational software will be valued at an estimated $500,000,000, a figure twice that of combined revenue from print and audio-visual instructional materials. The market for software alone, it is said, will accomplish in four years what it took the market for print material 150 years to achieve (Arachtingi, 1982).

Clearly, critical decisions of educational policy-makers have been affected by the benefits thought to be associated with regular computer use: improved cognitive and metacognitive functioning (Papert, 1980; Sheil, 1980) and/or improved classroom performance (Levin, 1980). Across the country, schools of education are integrating computer-literacy requirements into coursework requisite for graduation (Electronic Learning, 1982). Strong support for their commitment has been shown by the Association of Supervision and Curriculum Development, which recently recommended that teachers be eligible for certification only if their professional preparation include "knowledge of the principles of technology" (Harkness, 1982, p. 4). An equally enthusiastic endorsement of computers in educational settings comes from France, whose government has made computer programming compulsory throughout the secondary school level. Promotion of this phenomenon shows no sign of abating, recent warnings notwithstanding: "...growth in the number of computer classrooms," reports Education Week (1982), "is significantly outpacing both the number of teachers trained to use them and the ability of education schools to incorporate the technology in terms of faculty expertise, coherent programs and basic research" (p. 16).

An informal nationwide survey conducted for the French Ministry of Communication (1982) in this country states repeatedly that ineffective teacher-training is posing a serious obstacle to effective classroom use of computer capabilities. Teachers themselves have characterized much of their preparation inadequate (Sheingold et. al., 1981), citing, specifically, shortage of time to plan and actualize plans. Arachtingi (1982) suggests that although often not initially not hostile to the introduction and integration of microcomputers, teachers not sufficiently well-prepared by professional training programs become increasingly so (p. 14). Having no idea how the processes and products of the innovation should appear when implemented, teachers restrict or abandon their use of it (Gross et. al., 1971). This theory is illustrated by Kenneth Komoski's (in the New York Times, 1982) description of teaching equipment from the 1960's and 70's, now gathering dust in school storerooms: "(it was) oversold as an instant cure for educational programs and...not backed up by... proper adaption for school use" (p. 36).

Conditions and recommendations.

Diagnosis/Entry. Investigations of over 3,000 high schools (Aslin and DeArman, 1976; Cowelti, 1967) conclude that innovations often abandoned are those thought by users to be "complex," whereas those described as "comparatively simple" are generally retained (cf. Rogers, 1962). It is suggested, therefore, that trainers act to convince teachers that the innovation can be of immediate, specific use even in the absence of expertise. One means of accomplishing this is to target training toward a match with participants' idiosyncratic needs for information and assistance. Determination of these needs must not rest with administrators, as school leadership has been shown to "frequently adopt a view of teacher functioning near its own center of control" (Baldridge, 1975, p. 133). The judgment of consultants and other external agents is often equally constrained by special interests and skills (Baldridge, 1975, p. 141). Finally, teachers' self-reports are similarly open to narrowness and subjectivity, such that it is often suggested that trainers seek multiple measures of teachers' needs. These should be balanced between teachers' and others' perceptions, and may be secured from a range of observations (naturalistic, behavioral, technologically-supported, stimulated recall; conducted by peers, supervisors, advisors, administrators, trainers and/or consultants); anecdotal reports (interviews, ethnography, questionnaires, surveys, checklists, investigative/adversarial interrogation); and artifactual reports (review of policy documents, examination of student work and records, examination of classroom and school environment). The goal is, of course, to obtain an accurate assessment of the attitudinal, contextual and tactical attributes which influence the success of interventions. Excell (1982), for example, has documented the failure of computer training inconsistent with assessments of teachers' capabilities and concerns. Following the investigations of Maslow (1954), Fuller (1959) and Hall et. al. (1975), she suggests that newcomers to an innovation are most apprehensive about the direct impact of its demands upon them. If not articulated and addressed, teachers' feelings of inadequacy inhibit progress toward "higher-level" concerns: classroom management; student outcomes; collaboration with peers; and adaption of the innovation to novel situations (cf. Hall, 1978).

Training/Implementation. Once data have been collected which inform a school's selection of, understanding of and commitment to a set of short and long-term goals, systems of delivering training toward these goals must be established. Context features (identification of participants; group size, materials; mechanical aids; identification of instructors) and interaction patterns (lecture/recitation; modeling/roleplay; teamwork; tutoring/coaching; discrimination training; microteaching; experiential learning) come under consideration. Generally speaking, successful training employs motivational elements, such that participants are able to make attributions of achievement to themselves. In the case of computer training, rewards are experienced by means of rapid familiarization with features integral to standard use: the keyboard, disk drive, disks, game paddles/joysticks. Ideally, this is done pursuant to teachers' running of programs pertinent to their interests; by the close of the initial training session, one or more pieces of software should be known and immediately applicable to classroom settings. This is equally true in the case of influential school administrators (cf. Fullan and Pomfret, 1977), whose endorsement of training may come in the course of experimentation with easily-managed tool software. In all instances, usefulness is greatly enhanced by support materials and suggestions for preparatory or follow-up activity (Fullan and Pomfret, 1977).

Instruction of programming at the introductory stages is to be avoided; above all else, it is complex. Conceptual understanding of programming is founded on interconnections of situation-specific knowledge fragments (Atwood and Ramsey, 1978) and component rules (Brooks, 1977). Serious errors and much misunderstanding are not only likely but characteristic of novice programmers (Jeffries, 1982; Sheil, 1980). In fact, Nansen (1982) attributes the success if his teacher-training computer course to the initial avoidance of instruction in programming.

Evaluation. Typically, program evaluation has little effect on the learning process: information gathered is generally unavailable to end-users; irrelevant questions are asked; the whole, rather than the interactive parts, of training procedures are investigated; and the evaluative process, most often summative in nature, extends beyond the life of the training and cannot inform it (Cronbach, 1982). Curbing this waste requires careful advance planning, such that standards for success are set and agreed upon prior to training (AERA *Pre-Post Press*, 1982); a variety of instruments to document ongoing procedures is selected and pilot-tested prior to training (Eisner, 1981); and a multiplicity of behavioral and attitudinal results is expected and sought in target trainees, other teachers and students.

References

Anderson, C. Technology in American Education 1650-1900. Washington, D.C.; Technological Development Project of the National Education Association, 1962. (ERIC ED003747)

Arachtingi, J. Les Nouvelles Technologies de Communication et l'Enseignement dans les Etablissements Scolaires Americains. Paris: Centre d'Etudes d'Opinion, 1982.

Aslin, N. and DeArman, J. "Adoption and abandonment of innovative practices in high schools." Educational Leadership, 1976, 22, 601-06.

Atwood, M. E. and Ramsey, H. R. Cognitive Structures in the Comprehension and Memory of Computer Programs: An Investigation of Computer Debugging (Technical Report No. TR-78A21). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences, 1978.

Baldridge, V. "Organizational change and the consultant's role; rules for effective action." in V. Baldridge and T. Deal, Managing Change in Educational Organizations. Berkeley: McCutchan Publishing Corporation, 1975, 132-144.

Brooks, R. E. "Towards a theory of the cognitive processes in computer programming." International Journal of Man-Machine Studies, 1977, 9, 737-751.

Cowelti, G. "Innovative Practices in high schools: who does what--and why--and how." Nation's Schools, 1967, 79, 44-49.

"Microprocessing revolution." Education Week, 1982, 33 (1), 16-19.

Eicholz, G. and Rogers, E. "Resistance to the adoption of audiovisual aids by elementary school teachers." in M. Miles (ed), Innovation in Education. New York: Teachers College Press, 1964.

Eisner, E. "On the differences between scientific and artistic approaches to qualitative research." Educational Researcher, 1981, 10, 5-11.

"EL survey shows: schools of education are beginning to require computer literacy." Electronic Learning, 1982, 2 (3), 10-13.

Excell, D. A. Measuring Teachers' Concerns during the Implementation Process of Computer-Assisted Instruction. Unpublished doctoral dissertation, Teachers College, Columbia University, 1982.

Fullan, M. and Pomfret, A. "Research on curriculum and instruction implementation." Review of Educational Research, 1977, 47, 335-397.

Fuller, F. F. "Concerns of teachers; a developmental conceptualization." American Educational Research Journal, 1969, 6 (2), 207-226.

Gross, N., Giacquinta, J. B. and Bernstein, M. Implementing Organizational Innovations. New York: Basic Books, 1971.

Hall, G. E. The Study of Teachers' Concerns and Consequent Implications for Staff Development. Austin; Research and Development Center for Teacher Education, University of Texas, 1978.

Hall, G. E., Loucks, S. F., Rutherford, W. L. and Newlove, B. W. "Levels of use of the innovation; a framework for analyzing innovation adoption." Journal of Teacher Education, 1975, 26 (1), 52-56.

Harkness, R. "Competency in media technology needed." Long Island Assocation for Supervision and Curriculum Development Newsletter, 1982, Winter, 4-7.

Hechinger, F. "Computer pitfalls spur school vigil." New York Times, December 7, 1982, p. 36.

Jeffries, R. A Comparison of the Debugging Behavior of Expert and Novice Programmers. Paper presented at the Annual Meeting of the American Educational Research Association, March 1982.

Levin, J. A. and Kareev, Y. Personal Computers and Education; Thr Challenge to Schools (CHIP Report No. 98). La Jolla, CA; Center for Human Information Processing, 1980.

Maslow, A. H. Motivation and Personality. New York: Harper and Row, 1954.

Nansen, C. "Teaching computer use—not programming." Electronic Learning, 1982, 2 (3), 24-31.

Papert, S. Mindstorms. New York: Basic Books, 1980.

"ERS oulines program evaluation standards." Pre-Post Press (Division H Newsletter, American Educational Research Association), 1982, 8 (1), 1-7.

Rogers, E. M. Diffusion of Innovations. New York: Free Press of Glencoe, 1962.

Sheil, B. A. "Teaching procedural literacy." Proceedings of ACM Annual Conference, 1980, pp. 125-26.

Sheingold, K., Kane, J., Endreweit, M., and Billings, K. Study of Issues Related to the Implementation of Computer Technology in Schools. Final Report, National Institute of Education, 1981.

COMPUTERS ARE HERE

Glen L. Hampshire
Director of Computer Science
Jefferson High School
Portland, Oregon

With a staff of six instructors, 24 terminals for student use, and a four year curriculum, the computer science program at Jefferson High School is one of the most comprehensive high school computer science programs in the nation. Computers are Here, a 17-minute video tape, documents this as well as other components of computer use at Jefferson High. The computer science component offers students the opportunity to become proficient in four computing languages. Beginning students learn about the potential and limitations of the computer while learning the BASIC language, while fourth-year students, who have studied FORTRAN, COBOL, and Pascal, design complex programs and receive on-the-job training. The video tape also explores the computer assisted instruction programs at Jefferson. Chapter I (low skills) students are pictured and the use of the Computer Curriculum Corporation material is explained. Another component, "word processing", is a unique dimension of the Jefferson curriculum. Here business education students, studying office procedures, obtain training on word processing systems in a model office setting. As a management tool, the computer is widely used at Jefferson. A variety of uses is shown as another component of this exemplary program. "Computers are here" and the Jefferson program, as documented in the video tape, has become a dynamic program. Many agree that it could serve as a national model in high school education computing.

The information that follows is divided into seven categories. They are: the Computer Science Program, related classes, use of computers for management, use of computers for assisting instruction, a training center for teachers, some national recognition, and a review of hardware.

Computer Science Program

At the start of the 1982-83 school year, the Computer Science Program consisted of over 270 students, six teachers and an aide, a four-year curriculum and access to a Honeywell Mainframe via 24 terminals and three printers.

Computer 1-2 is the first of four courses that comprise the four-year curriculum. The course is available for grades 9 through 12 and meets the Jefferson High School requirements for Computer Literacy as of September, 1983. All the students entering the course must be taking or have completed Algebra I. The text currently used is the second edition of The Mind Tool, by Neill Graham, copyright 1980, West Publishing Company. The length of this course is one year and students meet five 47-minute sessions each week. They are scheduled into the computer laboratory two times each week with the possibility of a third session each Friday. Students write, on the average, one program per week. They are required to produce two major programs during the year. All programs include appropriate documentation and structured programming is stressed. The current language used is BASIC. Debugging, flowcharting, and prediction of the results of program execution are common activities in Computer 1-2.

Computer 3-4 is the second year-long course in the computer science curriculum. The goal of the course is to produce independent programmers. All students entering the course must have a credit in Algebra I and have a grade of C or better in Computer 1-2 or have permission of the instructor. The text currently used is An Introduction to Programming and Problem Solving with Pascal, second edition, Schneider, Weingart and Perlman, copyright 1982, Wiley. Pascal was chosen because it demands more structure and precision and better augments the concepts of the course. Students class/lab. schedule is identical to Computer 1-2 students. In addition to the curricular material, students learn editing, printing, TSS and batch, internal and external documentation, and user friendly strategies. Students complete the course with a four-week project of their own design.

Computer 5-6, the third course in the computer science sequence, is also one year long. This course is available for those students who have successfully completed Computer 3-4. The texts currently used are An Introduction to Programming and Problem Solving with Pascal, (see Computer 3-4) and Assembly Language Programming for the Apple II, Mottola, Osborne/ McGraw-Hill. Class/lab. scheduling is identical to Computer 1-2 and 3-4. Students write programs in several languages: Pascal, FORTRAN, and 6502 Assembly. Lab activities, including major projects, reflect topics studied (such as data structures, searching and sorting, and analysis of algorithms).

Computer 7-8 is the fourth course of the computer science sequence. Students entering this course are required to have successfully completed Computer 5-6. Various texts are used in the course, which is one year long. Class/lab. scheduling is identical to previous courses for one-half year. During the other one-half year, students will have an on-the-job training experience at a company in the Portland area. Classroom activities are seminar-centered and include guest speakers and sharing of programming projects. Students write individual and group projects in any of several high-level languages (Pascal, FORTRAN, or COBOL).

Computer Literacy will be offered as a semester course during the 1983-84 school year for those students who either lack the skills to take

Computer 1-2 or are not serious about following the four year computer science sequence.

Related Courses

Data Entry is a unique offering at the high school level. The course prepares students with an entry-level skill as a "keypunch" operator. At Jefferson we use Mohawk key-to-tape equipment that was donated by Consolidated Freightways. The keyboard manipulations learned on this equipment are applicable to card punch or to the most modern key entry equipment. Students also have experience in entering data using the mainframe terminals and microcomputers. Students must have a prerequisite typing skill of 30 words per minute. They learn to input data from many different types of source documents (accounts receivable and accounts payable, invoices, payroll and timecard data, cash register tapes, name and address lables, and checks and deposit slips). The importance of accuracy as well as speed of keying data is emphasized and developed. The student recognizes the integration of accounting, recordkeeping, data entry, and data processing. The system software needed to interface the Mohawk equipment with the Honeywell Mainframe was written by a computer science student.

Legal Secretarial/Word Processing class prepares high school students for a career as a legal secretary. Because of the unique training, which includes word processing, graduating students have little difficulty finding employment in any office setting. The classes meet for two periods each day and students can spend two years in the course. The main concepts learned include legal office procedures, legal document typing, word processing and employment preparations. Learning activities include machine transcription, typing projects, exercises on the IBM Mag Card II, Royal 6000 D, and CPT 8000. The texts used in the course are: Legal Secretary's Complete Handbook, Second Edition, Besse May Miller, Prentice-Hall, Inc., Legal Office Procedures, Marjorie Dunlap Bate and Mary C. Casey, Gregg and Community College Division, McGraw-Hill Book Company. The supplementary materials include: Legal Transcription and Dictation (Western Tape Corporation) and Handbook for the Legal Secretary (McGraw-Hill).

Use of Computer for Management

Jefferson uses computers on various levels for management purposes. There exists the usual services of the Portland Public School system's data processing operation. The attendance, class period by class period, is now a central operation which originally was maintained at the high school level. Beyond this, the library makes use of an Apple II 48K microcomputer to generate overdue notices and bibliographies for the library, and many instructors use the time-share system to replace the typical teacher grade book. The variety of uses is determined by the expertise of the individual teacher. One teacher, for example, has a lesson plan available for substitutes at the "touch of a few buttons."

The Use of Computers for Assisting Instruction

Jefferson High School is the site for a Computer Curriculum Corporation C.P.U. The CCC curricula is used in conjunction with our Chapter I (formerly Title I) reading and math program. Math students spend about 15 minutes per school period working on reviewing and developing elementary math skills and in problem solving. Selected students spend 10 minutes each day using basic reading programs. CCC is used in addition to the regular Chapter I curriculum. Another facet of assisted instruction is the use of microcomputer programs by teachers at Jefferson. Our library has a microcomputer and an ever-growing menu of courseware for instructional purposes. Teachers have a ready mechanism to evaluate microcourseware. And--Jefferson teachers have access to a large C.A.I. library via the time share terminals. Unfortunately, time available for students to use this source of materials is minimal due to the expanding computer science program. A catalog of these time-share programs, similar to an audio visual catalog, was produced in July, 1981, and teachers can use these materials with their classes when terminal time is available.

A Training Center for Teachers

The computer laboratory is busy from 4:00 p.m. to 9:30 p.m. on Tuesday and Thursday nights and 4:00 p.m. to 7:00 p.m on Wednesday nights as adult students are given an opportunity to learn about computers, available courseware and programming. Computers in Education I students are introduced to the limitations and general uses of computers as tools to aid instruction. They learn to review courseware on both time-share and microcomputers, and they experience CCC curriculum in a laboratory setting. They also acquire a beginning knowledge of BASIC and write simple programs during this initial course. Students enrolled in Computers in Education II are given the opportunity to build a useful program or system of programs to meet their "on the job" needs. The third course has been recently changed to focus on LOGO as a beginning programming language. Over 300 district staff members have been exposed to computing through these courses since the fall of 1980. All three courses offer both graduate and undergraduate credit through Portland State University.

Some National Recognition

During the 1982-83 school year, Jefferson High School was nominated for and received one of fifty national $20,000 grants from the Ford Foundation. The proposal that was submitted focused on computer literacy for all students, staff and interested community adults. The staff was then given opportunity to participate in one or more of twelve offerings from January 13th through March 18th. Each course was designed to encourage hands-on time to insure participants more than an abstract knowledge of microprocessor applications. In addition, a three-hour morning class was offered to forty-eight community adults during the same time interval. As the 1982-83 school year progresses, a comprehensive report will be submitted to both the Superintendent of Portland Public Schools and the Ford Foundation.

A Review of Hardware

Twenty-four terminals at Jefferson High School are connected to the Portland School District's Honeywell Level 66 Central Processor. With appropriate permission, students can execute professional level programs which handle large files. The P.P.S. computer system has well over one billion bytes of secondary storage and three meg/byte of main memory. The Mohawk data entry system at Jefferson was used by Consolidated Freightways until the spring of 1980 to enter C.F. trucking data collected from points all across the U.S.A. C.F. is one of the largest "computer users" in the Northwest. The word processing equipment at Jefferson consists of one CPT 8000, one Royal 6000 D, and two IBM Mag Card II machines. The Computer Curriculum Corporation C.P.U. on site is a CCC Model 17, which can handle 97 ports and has 27 different courses available for users. Sixteen of those ports (terminals) are located at the Jefferson site with others at four other sites. Jefferson also has six "phone modem" terminals that can access the Honeywell. Six microcomputers are located throughout the building with a documented need for many more.

Summary

The staff at Jefferson, like so many across the United States is involved in an adventure with the "Mind Tool". The potential to help educate our nation's youth by using the computer and other equipment made possible by the mighty microprocessor at this point in time seems boundless. We are excited! We covet your suggestions! Come and join us in our adventure!

** A Jefferson High School Senior, who is enrolled in Legal Secretarial/Word Processing and Computer 1-2 was responsible for the production of this document, using a CPT 8000. Thank you, Pam West.

CAUTION! YOUR GARDEN OF EDEN HAS AN APPLE II

Dr. Dennis C. Hansen
Westside Community Schools
909 South 76 Street
Omaha, Nebraska 68114

The rapid expansion of computer technology has once again allowed our educational system to be lured down the path of shallow planning, narrow focus, short-sighted vision and mismanagement of straining budgets. The increasing demand of the public for an educational response to this explosion of technology has caused a wide-spread bandwagon effect to govern the implementation of questionable practices and computer use in the schools. Yet the basic concept of computer technology lends itself to an exciting, versatile and meaningful tool for the betterment of education and the learning process. Well conceived goals and strategies can reverse the momentum of comdemning the computer to the AV closet and place in the hands of students and teachers a positive power here-to-fore unequaled in mass education.

The opening question for discussion in a recent five state computer conference sectional was "How are computers affecting the learning of the students in your school system?" The uncomfortable silence from the 80 people in the room was broken by the next probing question, "How many of your schools have computers?" The response was hesitant but the final count included everyone in the room!

This scene is being played time and time again in countless gatherings of interested, curious and somewhat unsure educators across America. A most important and immediate concern seems to be rising between 'do you have the material goods' and 'are they affecting learners in a positive manner'.

For what might have been the first time in American educational history, a resource for teaching and learning was thrust upon the mass education community before they even knew it was in their neighborhood. To further compound the confusion that insued; the press came from students, patrons and the manufacturers rather than the conventional development/delivery process within the educational structure itself. The reactions of schools have been at best an exploratory experience. An isolated short course by an individual and willing teacher or a Skinnerian implementation in a basic skill area while someone in the schools figured out what the computer could really do to enhance the learning process.

Further uncertainty was injected by the media producing community which exploded in numbers and product volume when the panic of software needs was apparent. The issue of quality product and the best method of teaching a particular concept seemed trampled under the rush to produce a diskette that would be quickly purchased under the pressure of 'need to put something in the hands of students'.

Now that the heat of passion has given way to a first experience in the romance with the microcomputer, can a lasting friendship be developed and a lifelong marriage of viable learning and valuable electronic delivery systems be reasonably assured? The potential is excellent if the proven process of quality planning and progressive strategies are employed.

In many school systems some computer goods may be in place or at least purchased and put somewhere. The opportunity to explore and discover is basic to learning. The isolated and exploratory activities that have been done are not in vain. But beyond the exploration and awareness there is a wide frontier. How do we settle and grow?

Just as the implementation of a new idea in the business world would require a needs assessment so must the proposed use of computers in a specific school system undergo such study. Consideration must be given to the current methods used to deliver instruction to students and a look at the ways in which the student population learn. Can the computer enhance the present methods and learning styles? A study of all curriculum areas should be made; not to find where the computer can be squeezed into the process but rather where the computer can be used to provide a better opportunity for learning. How are students meeting the challenges of creativity, problem solving and real problems? Can the computer be used to develop a wider spectrum of alternatives for solution and better express ideas? The computer has shown great potential in assisting in the management of student progress. Are there specific areas in the school system where computer use can be employed to improve the instructional and learning situations? What are the direct and indirect needs of the community? Can the school assist in the betterment of the business and community life through the involvement of computers in the educational offerings? What are the desires of the students themselves as they prepare for their future? What role can the knowledge of computers and how to effectively use them play in the life planning of these students? Throughout the exploration of needs the questions can be consistently measured against the attributes of efficiency and effectiveness. If the target area cannot be improved by computer use the search should continue for other methods.

Though not always thought too important, a well conceived philosophical base must be developed around the needs and values of the school and community. The call for recognition of computers as a legitimate learning aid must be given. A clear statement of belief in how the computer will affect learning and the community must be made. Finally a response to the way the computer will affect the society of the community must be declared.

A well designed assessment of needs will provide the basis for development of goals for computer use in the schools. These goals should reflect the value placed on variety in learning methods and on the ways the schools can best impact the learning process. The issues of ethics, societal impact and discriminative use of computers should be covered in the set of goals. Finally, an evaluative process must be included as a goal.

The development of objectives must be done on a continuum across grade lines for the total system. A consistent review of the established goals can provide guidelines for the building of a set of general objectives. For example, the objectives may be designed by application areas, ie: Computer Literacy, Computer Assisted Learning, Computer Managed Learning, Simulation, Problem Solving and Programming. These applications must answer to one or more of the school system goals. The objectives then can be detailed and assigned to introductory levels on the total system compendium. Consideration should be given also when developing and assigning objectives to include staff knowledge needs and the needs of the adult population in the community. A broad coverage of people in the defined objectives lends support to a community based curriculum and thus a draw for wide support.

The effectiveness of an accepted and supported computer involved curriculum lies in the planning design and implementation. People need to be assured where they fit into the plan, how their skills will be affected, how the curriculum will be affected and a time frame when phases are to be accomplished.

Time lines are effective in demonstrating when expectations are to be met both from human resources and in budgeting considerations. They also show a totally planned conception of the use of computers in the school rather than a piece-meal, add-on approach.

The provision of adequate training for school staff and community people alike must be laid out in some detail. The acceptance level of computers as a learning tool can quickly become negative if people feel inadequately prepared and thus less effective in assisting the students in learning.

A crucial decision arises in trying to place the use of computers somewhere in the already filled school curriculum and school day. This issue probably takes the longest to decide and raises the most discussion. Again a reference to the system's goals and staff preparation will offer some guidelines. However, the objectives and how they support the general curriculum must be studied in depth. Note where the priorities lie. If computer assisted learning is to be given the main emphasis then curriculum task forces must assist in determining where the objectives fit. If programming or specific courses including problem solving are to be utilized then quite a different decision of where these areas are to be placed will need to be made. Though such decisions are unique to the school system, the standards should always be met. The decision is a total system one. The major purchase of support hardware/software should not be made until after a plan is in place.

Though budgetary considerations may well begin to focus on the dollar need from the very conception of planning, not until a well designed implementation and staffing plan and time line are proposed can the cost be considered. If after the plan is given the cost is deemed too much then the plan can be revised or redesigned from the goals through the implementation stage. The needs assessment is still valid and the philosophy remain intact. To start with a dollar amount and try to buy as much program as you can leaves too much to chance and too often disregards the very basis and foundation needed for quality planning.

The evaluation of the effectiveness of computer involved learning must be considered from the initial philosophical development through an adequate amount of time for the implementation to prove its value. Consistent checks need to be planned and made known in advance in order to chart the areas where change is necessary. Too many educational programs meet an early demise due to the lack of periodic checks and changes as the plan progresses.

A strength of the American educational community has been its willingness to meet the ever changing needs of people with a professional, creative, and beneficial response. Educators can regard the challenges by the technological age as a positive affirmation of their abilities to examine the alternatives presented and to utilize wisely the new tools for assisting the learner to grow. We must proceed with eagerness and confidence using the skills of proper planning and decision making that have served so well.

Creating Instructional Materials for the Microcomputer

Janet D. Hartman
University of Evansville
P.O. Box 329
Evansville, IN 47702

INTRODUCTION

The accessibility of the microcomputer as an instructional tool has increased the need for quality educational software. It is often difficult to find software which adheres to the principles of good instructional design. Development of quality educational software involves a top down, three-phase process: (1) design, (2) writing and (3) implementation/evaluation. Most often a content specialist carries out phases one and three and determines the guidelines to be followed in the writing phase. The programming component, however, may or may not be carried out by the author. The outcome of the instructional and technical teamwork is a piece of quality software.

THE DESIGN PHASE

The initial step in the design phase is a feasibility analysis. This includes the selection of a topic and an assessment of its appropriateness for computer application. An assessment of the students who will use the program must also be made to determine their ability levels, reading levels and any other special characteristics which they may possess.

Once the topic has been deemed appropriate and the audience has been identified, the mode of instruction should be selected (tutorial, simulation,drill and practice). This is followed by a written statement of the rationale, an identification of subtopics and a list of specific objectives.

The development of subtopics assists in determining the exact content to be covered, the sequencing of material and facilitates the identification of specific tasks within each subtopic. A subsequent analysis of each learning task aids in pinpointing the concepts to be learned and provides a framework for the hierarchy in which the tasks fall. A great deal about the content of the proposed module can be learned through the task analysis technique.

Once an analysis of all tasks has been performed and the relationships between them have been determined, specific objectives should be written. These should be stated in terms of student outcomes, that is, they should be behaviorally stated.

After the elements which will make up the module have been represented behaviorally and a progressive sequence through the material has been selected, it is essential to design the overall system and represent it in some symbolic manner. This may be a flow chart, a structure chart or some other graphic choice. A system chart which represents the sequence of the subtopics with sufficient detail to see the flow of concepts within each provides a tool for initial development as well as subsequent revision.

The systematic procedure described above produces a complete structured design from which the content module can be written. In the writing phase the actual text and graphics which amplify the design are produced.

THE WRITING PHASE

Writing the module from the completed design is the second phase of the project. An actual layout, screen by screen, with directions for feedback and branching speeds up the process of programming considerably. In some specific cases the layout of text in accordance with screen dimensions can reduce the time spent debugging the program once it is produced.

Good instructional software should be "user friendly". To meet this objective the module should be a turnkey system so the novice user need only get the machine started to initiate the program. It is wise to make the program menu-driven if several choices of topics are available to the user. Users should be able to return to the menu at any point which they choose.

Good directions are essential to the success of the program. These should include instructions for using a HELP function, escaping the program, returning to the menu, responding to questions and using the program. Complete directions should be written after completion of the content frames so that specific items, such as the use of special characters, can also be considered. The power of the instructions to make or break the effectiveness of the module should not be underestimated.

Frames which include any text should consist of short sentences to increase the readability. The user should be able to follow the logic of the material presented with ease; transitions between concepts should be made smoothly. A limited amount of new material should be presented in each frame. Logical steps through the content should be small. All of these principles enhance the readability of the module.

Another guideline to follow is to limit the amount of text presented on a single screen. Appropriate spacing should be utilized to break the text into smaller, more easily read segments. Screens with too much text are difficult to read and increase the frustration level of the user. Questions should be clearly separated from responses and feedback. It is also helpful to have all three appear on the screen simultaneously so the user can relate his response to the feedback and the original question.

Frames should be student controlled unless the material is designed to be time driven. The instructions for proceeding from frame to frame should be clearly stated. This provides for self-pacing and various reading abilities. Scrolling of the text on the screen should be avoided.

The "bells and whistles" which make the personal computer so unique should be used sparingly. This includes flashing, inverse video and sound. The best approach to take is to determine whether their use enhances the content presentation. If not, avoid using them.

Limiting responses to questions to single character input provides for the poor or untrained typist. Open ended questions which require accurate spelling in the response are poor candidates for computer based materials. A list of responses limiting choices and requiring a single key response can be handled exceptionally well.

A summarization of material covered or some other method of closure should be utilized to indicate the end of the module or a segment of the module. With drill and practice units or testing units it is desirable to provide quantitative feedback as to the performance level of the user.

Programming the module is the final step in the writing phase. A consideration of possible authoring languages may be made at this point. As a part of the programming step any user materials which are not computerized should be developed as well. Adequate documentation of the program is essential.

IMPLEMENTATION AND EVALUTATION

No matter how well-written the final product is, it is likely to require some modification. Pilot testing with users should delineate the areas where changes need to be made, either instructional or technical.

An overall assessment of the module's ability to meet the stated objectives should be made based on student use and the author's professional judgment. A general checklist of items related to content quality, instructional quality and technical quality should also be utilized to identify areas of weakness and needed change.

CONCLUSION

The final product is a tribute to the author. The time invested in development of educational software is substantial. A finished product of high quality can provide a great service to educators and, ultimately, the time invested can be re-gained through usage in the classroom.

Programming Plans as Advance Organizers and Their Use in Improving Programmer Debugging Performance

Johnette Hassell *
Tulane University

Janet H. Lind +
Hewlett Packard Co.

Program maintenance represents a large (approximately two-thirds) part of the cost of software. A major contribution to this cost is the debugging process. As a method of improving performance in this facet of maintenance, programming plans, advance organizers and their relationship to debugging are considered. An experimental investigation into the relationships among these concepts is also outlined.

* INTRODUCTION

Throughout the lifetime of a software product the maintenance process accounts for approximately two-thirds of its entire cost [BOE76]. In addition, maintenance, an unpopular task usually assigned to the most junior programmers, is notoriously error-prone. As a result, much of software engineering has as its goal the reduction of software maintenance cost.

Software maintenance falls into the two broad categories of modification and debugging. It is the debugging process which is the focus of this paper. In the examination of a program for an error, a study of both program constructs and variable usage must be performed. These features are interleaved, and usage of a variable may directly influence the control structures in which the variable appears. Any previous information which the debugger has about the relationships between variable usage and control structures may be used in the debugging process.

One mechanism for mentally encoding such prior knowledge is programming plans, or scripts. Knowledge of appropriate plans for various types of programming activities can be used as an advance organizer by a person involved in debugging. This paper discusses advance organizers in general, the concept of scripts as a type of advance organizer, and the use of scripts in the programming and debugging processes. Also presented is an outline of an experimental investigation into these concepts potential for improving debugging performance.

* DEBUGGING

Debugging is a process which begins with some indication of the existence of an error, and requires determining the exact nature and location of an error [MEY79]. Meyers identifies three debugging techniques. One common technique is the brute force method. This method requires the least amount of thought and may be the most inefficient. One of the debugging techniques which falls into this category is debugging from a storage dump, which contains much irrelevant information and which was probably not produced at the exact time that the bug occurred. Another brute force method is the scattering of print statements through the program. This is a somewhat hit-or-miss method, and since inserting print statements requires program modification, additional errors may be induced.

* This research is supported by the National Science Foundation under MCS Grant No. 8103617. Travel support is provided by the Haliburton Foundation, Inc.

+ Janet Lind is a doctoral candidate in Computer Science at Tulane University.

Another more successful debugging method uses the induction process, in which one proceeds from error symptoms and their relationships to the error. The steps in this process require first locating pertinent data about both correct and incorrect program results. This data is next organized to allow patterns or contradictions to be observed. An hypothesis about the error(s) can now be developed and tested. This test may involve verifying that the hypothesis completely explains the clues or may require executing additional sample cases to determine that the results match those predicted by the hypothesis.

Debugging by the deduction process starts with a general set of theories and, via an elimination procedure, arrives at the location of the error. The first step in this method is the enumeration of the hypotheses about the possible error causes. Through these hypotheses, the available data can be analyzed. Various schema are used to select an hypothesis and the selected hypothesis is refined to correspond to a program section. A test is then attempted.

Note that both the inductive and deductive debugging processes require the formation of hypotheses. The hypotheses which the debugger can make are based on previous experience with, and knowledge about, programs. Two types of hypotheses are formed. For a given problem, a debugger can hypothesize about appropriate program solutions [MAG82]. The behavior of an errant program allows the formation of a second set of hypotheses about where the given program does not match the anticipated program solution. The following section discusses advance organizers which are learned as general concepts and aid in such hypothesis formation.

* ADVANCE ORGANIZERS

The debugging process requires learning. At the very least, the programmer must learn the "mechanics" of particular parts of the program. In addition, new programming techniques and even the problem itself may have to be learned. Thus, it is appropriate to consider how the debugger can use general concepts to learn about the program at hand. Ausubel describes a model for verbal learning using advance organizers. He identifies an individual's cognitive structures as knowledge of a particular subject matter at a given time and how well organized, clear, and stable that knowledge is [AUS63]. He maintains that a person's cognitive structure is the foremost factor governing how well new material is learned.

In the advance organizer model, learners are presented with an abstract concept before the learning experience. This reinforces, clarifies, and stabilizes the learner's current cognitive structure and provides the "intellectual scaffolding" for assimilating new material. Advance organizers are usually generalizations, major concepts, propositions, or laws of disciplines. For example, the general concept of culture can be used to organize factual information about various cultural groups. It is important to note that advance organizers are not merely introductory overviews, but rather, are major concepts with a higher level of abstraction and generality than the material to be learned.

A number of studies have supported Ausubel's theory. For example, student recall of ambiguous technical passages was improved when organizing material, such as title, topic sentence, or diagram, was presented before the passages [BRA72, DOO71]. Although several advance organizers have been studied, the one of interest to the authors has been studied under the topic of text understanding through the use of scripts.

* SCRIPTS

In text, large amounts of information are often left out, thus requiring an individual to have other knowledge of the situation and its details. These unmentioned ideas are actually organized around standard script situations [SCH77]. A *script* is organized knowledge needed to understand an event and may be used to point out appropriate behavior for a situation, such as how to act in a restaurant. A script encodes ordinary details in a chain that is essentially entered into the understanding process as a block. This block is called up whenever the script is activated. An early study of scripts, using master and nonmaster chess players, was conducted by Chase and Simon [CHA73]. The masters were found to be better at remembering the board positions of pieces when those pieces had been arranged in some meaningful chess configuration. Both the masters and the nonmasters did equally well at remembering random configurations. It appeared that for a meaningful configuration, the masters used a higher level knowledge to organize the board into easily remembered pieces.

In a discussion of dinner at an exclusive restaurant, some acts may be omitted. Knowledge of an appropriate script allows one to assume that reservations were made and the bill was paid. Scripts are used to supply missing, but important, details and if too many script steps are omitted, accurate script selection cannot be made. Scripts are hierarchically organized and in a higher level script, a single step may be expandable as a script on its own. Deviations from a script may have repercussions in later understanding, for it may be possible to know that some condition exists which violates successful completion of a script. For example, knowing that a person´s wallet was lost may give a clue that a restaurant script cannot be completed normally.

In applying a script, it may be necessary to make different types of inferences. As previously mentioned, it is necessary to infer the existence of intervening events which must have taken place between two explicitly mentioned events. Another form of inference matches specific elements of the text to variables in the script. Finally, it is also necessary to recognize multiple references to the same specific role filler.

Scripts, then, are examples of cognitive structures which an individual can use in understanding and categorizing material that must be learned. For example, if one wishes to dine at an exclusive restaurant in a foreign country, knowledge of restaurant scripts will lead to inquiries about obtaining reservations, transportation, and what form of payment is acceptable.

As the restaurant example shows, knowledge of a script can lead to the asking of questions that solve a problem or provide information for dealing with a situation. Thus the script, which represents previous knowledge, serves as a mechanism for aquiring new information and acts as an advance organizer. Although most of Ausubel´s work deals with verbal learning, i.e., learning from verbally presented material, advance organizers need not be so restricted, e.g. [MAY81]. The concepts of advance orgaizers and scripts exist in programming and, the authors believe, can be used to enhance debugging activities. The following sections develop these ideas and present an experiment to investigate the relationship between scripts as advance organizers and debugging tasks.

* SCRIPTS IN PROGRAMMING

In a study of the learning techniques and cognitive methods applied in the programming process, Soloway, Ehrlich, and Bonar describe the use of "plans," which are very similar to "scripts" [SOL82]. For clarity, the word "script" will be used in describing the work of Soloway *et al*.

Two types of scripting are presented. In one view [SOL82], scripts are organized around the roles that variables play. The script knowledge in programming is actually a catalog of stereotypic actions sequences, which are similar to the scripts people use in understanding stories. In one study, similar to Chases´s chess player experiment, programming experts could remember programs better than novices when the programs had meaningful structures. The belief is that experts use higher level knowledge, or scripts, in order to code the presented programs for easier recall [SHN76].

Consider the following two algorithm fragments:

Algorithm A

```
(1) Count <-- 1
(2) do-until end of input
(3)       read ARRAY [COUNT]
(4)       COUNT <-- COUNT + 1
(5) end do-until
(6) COUNT <-- COUNT - 1
```

Algorithm B

```
(1) COUNT <-- 0
(2) do-until end of input
(3)       COUNT <-- COUNT + 1
(4)       read ARRAY [COUNT]
(5) end do-until
```

Assume that the amount of input will never exceed the limit of the size of the array. When these fragments are executed, both routines will read values into ARRAY and at the end of each of the routines, COUNT will contain the number of items read.

In Algorithm A, line 1 (COUNT <--1) calls forth a script that requires that COUNT be incremented after it is used. In contrast, (in Algorithm B) COUNT <--0 initiates a different script in which COUNT is incremented before it is used. Thus, the selection of a script is triggered by the usage of a variable.

In an earlier perspective, Soloway et al. categorize scripts as they relate to problem types [SOL80]. In this view, a classification technique is used to group problems based on common features. For example, there are scripts for file updating, running computations, and text handling. In these instances the problem type triggers the selection of a script. Soloway and Woolf extended their work to advocate that a script can be used to develop programming templates which organize common and different features of programs.

In the debugging process, the debugger may attempt to match a known script against the current faulty program. A mismatch from the known script may point to the error site. Scripts may be examined forward from the point at which the script began to the error point or to the script completion. Scripts may also be examined backwards from the current state. The forward mode is more often used with scripts based on problem type. The backtracking method is used when one is working from a known item value and is looking for the previous steps which affected the value.

Weiser has studied debugging methods, and proposes that an individual breaks large programs into smaller, coherent pieces which may or may not be textually contiguous [WEI82]. This technique serves to reduce the cognitive distraction of unrelated program statements. Tracing backwards from a variable in a particular statement to identify all possible sources of influence on the value of that variable often reveals that many statements in the program have no influence at all. The process of stripping a program of those statements which do not influence the variable of interest is called program slicing. Slices are most useful for understanding a program when they are considerably smaller than the original program. Knowledge of scripts provides a directed, structured method for slicing.

Debugging is a cognitively demanding task requiring categorization, relationship recognition, and hypothesis generation and testing. The existence and use of programming scripts represents debuggers' attempts to reduce this cognitive strain. As experience increases and more examples of problems and programs are studied, additional higher level scripts evolve. These high level scripts are abstract representations of problem solutions and, as such, function as advance organizers.

* EXPERIMENTAL INVESTIGATION

If programming scripts do indeed serve as advance organizers for programs, then a debugger who has conscious knowledge of the scripts used in a program should perform better at debugging tasks on that (and perhaps a similar) program. The following experiment, which investigates this hypothesis, is currently underway at Tulane. Full analysis of the data is in progress. Results will be available at the conference.

A group of 120 undergraduate and graduate computer science majors of varing level of experience serve as the sample. Half were given programs to study with only the instructions "study this program and try to understand it." The other half were given the same program plus questions to draw their attention to the script usage. Both groups studied the programs for the same amount of time.

The study programs were removed and students were given copies of bugged programs and sample I/O. They were instructed to locate, but not correct, the bug(s). Half of the students from each group received the same program they had studied. The remainder received programs different from the ones they studied but which used comparable scripts.

It is expected that those students who studied via scripts and debugged the same program will perform better than the other three groups. Those with unstructured study and a different program to debug are expected to show the poorest performance. The relative performance of the remaining two groups is to be determined.

* CONCLUSION

Advance organizers are abstractions that assist in the learning of new material. In programming and debugging, one form of advance organizer is scripts. These programming scripts allow the debugger to anticipate a program solution to a problem and to use the hypothesized solution as a guide for finding probable error sites in the program. Investigation is underway to determine if conscious knowledge of the scripts used in a program will assist the debugger in performing the debugging task. It is expected that the structured preview provided by script recognition will significantly improve the performance of subjects on debugging tasks. Full statistical data will be available at the AEDS conference.

BIBLIOGRAPHY

AUS63 Ausbel, David, The Psychology of Meaningful Verbal Learning, Grune & Stratton, Inc., New York, 1963.

BAR71 Barron, R., "The Effects of Advance Organizers Upon the Reception, Learning, and Retention General Science Concept," DHEW Project No. 1-030, ERIC Document Reproduction Service, ED 061554, November, 1971.

BOE76 Boehm, B. W., "Software Engineering," IEEE Transactions on Software Engineering, C-25 (12), December, 1976.

BRA72 Bransford, J. D., and M. K. Johnson, "Contextual Prerequisites for Understanding: Some Investigations of Comprehension and Recall," J. Verbal Learning and Verbal Behavior, 11, 1972.

CHA73 Chase, W. G., and H. A. Simon, "Perception in Chess," Cognitive Psychology, 4, 1973.

CLA73 Clawson, W. and B. Barnes, "The Effects of Organizers on the Learning of Structured Anthropology Materials in the Elementary Grades," Journal of Experimental Education, 42, 1973.

DOO71 Dooling, D. J., and R. Lachman, "Effects of Comprehension on the Retention of Prose," Journal of Experimental Psychology, 88, 1971.

MAG82 Magel, K., "A Theory of Small Program Complexity," ACM SIGPLAN Notices, 17 (3), March, 1982.

MAY81 Mayer, R. E., "The Psychology of How Novices Learn Computer Programming," ACM Computing Surveys, 13 (1), 1981.

SCH77 Schank, R. C. and R. Abelson, Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

SHN76 Shneiderman, B., "Exploratory Experiments in Programmer Behavior," International Journal of Computer and Information Sciences, 5 (2), 1976.

SOL80 Soloway, E. and B. Woolf, "Problems, Plans and Programs," ACM SIGCSE Bulletin, 12 (1), February, 1980.

SOL82 Soloway, E., K. Ehrlich, and J. Bonar, "Tapping Into Tacit Programming Knowledge," Proceedings of Human Factors in Computer Systems, Gaithersburg, Maryland, 1982.

WEI82 Weiser, M., "Programmers Use Slices When Debugging," Communications of the ACM, 25 (7), July, 1982.

THE INFORMATION CENTER - END USER COMPUTING IN HIGHER EDUCATION A NEW DIRECTION

Barry R. Hemphill
Illinois Board of Governors Cooperative Computer Center
190 Prospect Ave.
Elmhurst, Illinois 60126

This paper will describe in comprehensive, overview form, the successful implementation of an Information Center in a multi-university service center environment. The Illinois Board of Governors Cooperative Computer Center serves three public universities in the metropolitan Chicago area.

INTRODUCTION:

The need for rapid access to data in support of administrative operations at colleges and universities is ever increasing. Demand placed on computing facilities increases at an almost geometric rate. This problem is exacerbated by the budget constraints placed on most of higher education today. The conventional approach to service delivery in data processing organizations has been inadequate and will be increasingly so in the 80's. One approach which has exceptional possibilities for increasing and improving services while maintaining a level data center staff is the Information Center. The heart of the Information Center is a small staff of trained data processing professionals dedicated to providing end-user data processing services. The tools are an array of user-friendly software packages to facilitate end-user access to computing resources.

ENVIRONMENT:

The Illinois Board of Governors Cooperative Computer Center, located in Elmhurst, Illinois, was created by an act of the legislature in 1971 to take advantage of the inherent economies of scale provided by a computing consortium. The Cooperative Computer Center now provides a full range of administrative (and partial academic) computing services to three public universities in the metropolitan Chicago area. The universities served are Chicago State University, Northeastern Illinois University, and Governors State University, with an aggregate enrollment of 23,000 students. All three universities are urban commuter schools with both baccalaureate and masters degree programs.

WHY AN INFORMATION CENTER?

Most data processing organizations are faced with some, if not all, of the following "challenges":

- An ever-increasing demand for "maintenance" types of service.
- Outside usage of time-sharing services.
- Pent-up demand for micros, PC's, etc.
- Increased personnel costs.
- A desire on the part of the users to utilize decision support systems to provide better, more varied information to middle and upper management.
- A requirement for fast access to information.
- Tight budgets, coupled with demand for increased services.
- Requirements for unscheduled, on-demand access to data.
- An increasing backlog of service requests.

While examining various options available to find a way out of the lose-lose environment described above, the Cooperative Computer Center determined that the Information Center provided a realistic path to satisfying many user requirements. After extensive discussion, meetings, and education, it was determined to begin implementation of the Information Center to serve the three universities supported by the Cooperative Computer Center.

THE APPROACH:

The mission of the Information Center is to provide the end-user community with a wide assortment of user-friendly tools to allow them the broadest possible access to university data. Additionally, the tools should be powerful enough to support reasonably complex analysis of data. The goal is to provide increased self-sufficiency of end-users in the arena of decision-support systems. The Cooperative Computer Center version of the Information Center will provide powerful capabilities for query and information retrieval, report formatting, graphics, text and word processing, and electronic mail. The Information Center is organizationally placed on the same level as applications programming groups. By structurally being set up as a separate unit, the Information Center personnel can concentrate on their mission and goals rather than becoming involved in the day-to-day problems of applications programming.

STAFFING:

The Information Center should begin with supervisor/manager and a staff of one or two. This group, ideally, will find their own areas of specialty within the product offerings of the Information Center. It is recommended that a specific and limited subset of products be supported. An unplanned smorgasbord will inevitably lead the program down the path to failure. The Information Center manager should possess strong, inter-personal skills and have a reasonably technical background.

SERVICES PROVIDED BY THE INFORMATION CENTER:

The Information Center provides specific services that are essential to the success of the facility. It is not a workable approach to simply provide products, manuals, and say to the users "here it is, have at it, do your end-user computing". Services provided are:

- Education
- Consulting
- Direct Assistance - it is important to note that one of the hallmarks of the Information Center is assistance. The information Center consultant should not do the job for the user. The consultant should assist. Otherwise the limited resources would inevitably be soaked up in a small subset of the user community.
- Newsletter
- User Manuals
- Technical Support
- Data Control
- Marketing
- Product Evaluation

HOW TO BEGIN:

Most successful Information Centers follow a similar track in their initial implementation. A pilot application should be selected. The application should be well-defined, and limited in scope, with sufficient pay-back to the user to make it obvious how using this powerful tool will improve his/her working environment.

When setting up the pilot, the following tasks should be accomplished:

- Select the product and be sure that the Information Center personnel are conversant with it.
- Select a pilot user who is interested and enthusiastic.
- Assign staff.
- Define the resources required.
- Develop education and training.
- Monitor the pilot.
- Prepare, in conjunction with the user, a presentation/summary at the end of the pilot period. The pilot consumes a relatively short period of time, one to two months.
- Modify approach, if necessary; continue implementation in other areas.

PRODUCTS:

At the Cooperative Computer Center, the following products were selected for the initial offerings. At the present time, constrained budgets have slowed the implementation, but not changed the overall plan. The products are all IBM software products as it was felt by Center management that the range of product offerings should have consistency and compatability from product to product so that data could be transferred easily from one to another.

- Query By Example (QBE) - Query by Example is a powerful inquiry facility which operates on data files either created by the end-user or extracted from the university data bases.

- A Departmental Reporting System (ADRS II) - ADRS II is an interactive package for creating reports from either external data files or data entered by the user at the terminal.

- ADRS II - Business Graphics - This package allows the business professional to request ADRS data to be displayed in a variety of graphics formats. The result is an easy to use facility for the production of colored graphs, reports, data, and texts.

- Document Composition Facility (DCF) - DCF is a text formatting program designed to process documents to be printed or displayed. It provides an extensive set of document formatting capabilities.

- The Professional Office System (PROFS) - The Professional Office System is aimed at increasing the productivity of managers, executives, and professionals.

The Cooperative Computer Center has implemented pilots in three areas: query, reporting, and graphics. Additional planned applications include the utilization of document formatting tools and professional/managerial office time management.

SUMMARY:

The Cooperative Computer Center's initial implementation of the Information Center concept indicates that the basic premise is sound. Higher education with its diverse users of data and complex organizational structures, coupled with the many interactions that are required with various internal and external constituencies, is a particularly viable candidate for the Information Center. Areas of universities that have traditionally had little data processing support are now able to receive that support. It is also apparent that as the Information Center tools are spread among the campuses, the requests for maintenance type data processing services will substantially decrease. Professional systems analysts and programmers are freed for new development and major changes to the large, transaction-based systems that support the university's data bases. The Information Center is not a new concept. Data processors have been attempting (with limited success) to use some of the cruder tools available in the 60's and 70's. These rudimentary tools, however, had many limitations. They lacked security, were cumbersome to use, and in most cases required a quasi-data processor to be effective. Whether we care to admit it or not, end-users in the 80's will be doing more and more of their own computing. The Information Center allows an organization to effectively focus scarce resources rather than ineffectively dissipate those resources in a smorgasbord of equipment and technology.

MICRO-COMPUTERS: PSYCHOLOGICAL REPORTING, DATA BASE MANAGEMENT, 94-142 IMPLEMENTATION

R. Keith Hoyle
Psychologist

Fairfield County Schools
Post Office Drawer 622
Winnsboro, South Carolina 29180

Fairfield County Schools are using the micro-computer to facilitate psychological report writing, due process procedures and P.L. 94-142 implementation. Benefits of this system include: less paper work, freedom for psychologists to participate in activities other than testing, and others.

Many presentations on micro-computers explore the theoretical uses and applications of computer technology in an effort to create an awareness of the potential of the computer. The emphasis of this presentation will center around specific programs developed and/or used in South Carolina by the Fairfield County Schools.

The Special Services Department of the Fairfield County School District began to realize the potential for use of computers to solve some of the problems in management and service delivery. An inordinant amount of time seemed to be spent in writing psychological reports, compiling data for state and federal reports, keeping track of due process timelines, and other paperwork associated with the implementation of P.L. 94-142 standards and guidelines. A micro-computer system including three terminals, five eight-inch floppy disk drives, a 40 mega-byte fixed disk, and a Xerox printer were purchased at the close of the 1979-80 school year.

With the system we purchased a data-base management program to allow us to store all pertinent information for each student assigned to programs for the handicapped. This program allows us to compile student counts based on a variety of criteria, generate class rolls, monitor timelines relative to due process, update files more accurately, generate data for reports to the state department, federal government and school board. In addition to the data-base program, one of our psychologists has written a psychological report generator which can be tailored or adapted to the individual psychologist who uses the program. Another utility of the system is the word processor which can be used effectively in volume mailings, and in form construction and revision.

With the exception of the word processor, all programs are written in the C-Basic language which provides for compatability and merging of utilities.

As already stated, our district is using this micro-computer system to manage data, generate psychological reports, and to facilitate the handling of repetitive secretarial tasks. Ongoing or future plans for the system include merging the psychological report generator with the data-base management system, the writing of a program to generate IEP's including daily or weekly student learning activities, maintaining inventory, and providing an accounting system for federal and local funds.

This system has been implemented with very little technical expertise on the part of the Special Services staff. Problems which have been encountered include: availability of proven software; compatability of existing software; down-time due to inadequate system; maintainance; adaptation to computer languages; excessive expectation of computer capability and speed; development time; and apprehension on the part of staff in learning to operate the system.

The micro-computer system has had a very significant impact upon the total Special Services program. Specific benefits include; freedom for the psychologist to provide other services such as counseling, implementation of recommendations, and involvement in curricular development aimed at specific needs for exceptional students; accuracy in typing and report management; reduction in time spent on paperwork; easier access to data by specific variables; frees secretaries for other work; and better organization of the department as a whole.

SETTING UP A COMPUTER PROGRAM
Now that you have a computer, what are you going to do with it?

Robert W. Jackson
Microcomputer Educator, Whitby School
2 Andrews Road
Greenwich, CT 06830

Abstract: It would appear from much of the literature and especially from computer hardware salesmen that learning to operate a microcomputer and step up a computer program is the easiest thing to do. Reality sets in when the computer is delivered to your classroom and the children crowd around. There is a definite need for preparation and planning before the computer arrives and a great deal of creativity to fit it into an already crowded curriculum. I would like to point out some of the directions you can take and tell you of some of the problems I have encountered.

Background: For the past four years, I have been involved in setting up and running computer programs for private schools. As a consultant hired by the schools, I come in and assess the needs of the school, oversee the purchase of the equipment, and set up the computer program. Educating the teachers has always been a major priority, and yet this has proven to be the most difficult part of my job. To this end I have visited hundreds of teachers using computers and tried to find different approaches to using computers in education.

I have come to realize that there are four elements which lead to the successful implementation of the computer in the education setting. First and foremost is the willingness of the teacher(s) to accept the computer and to try to implement it within the curriculum. Second, there has to be a purpose or direction to the use of the computer. Third is access to the computer and fourth is the availablity of some good software.

Frequently, I am called to a school which already has a computer, but the administration feels that they are making little use of it. Most often these machines made their way into the school system through the energy of a parent or teacher who has left the school. Without a person to use, maintain and encourage the exploration of this new technology, no computer program can grow to full potential. I have seen some groups of teacher successfully set up and run a program, but again there is usually a single person in the leadership position nursing the computer program along.

Willingness:
Before teachers will accept a computer into the classroom, they are going to have to become comfortable with it, and before they commit themselves to using it they are going to have to discover for themselves that there is some advantage to using it over traditional methods. In order to accomplish these things, teachers are going to have to

commit the time to play with the computer, and there has to be some good software to play with.

I usually use a word game such as Word Challenger for the TRS-80 or a hangman game for some of the other computers. I have also found that typing programs such as Microsoft's Typing Tutor are excellent programs to demonstrate to teachers the advantage of a computer. The value of a game at this stage is to reduce the image of the computer as a math machine, and the value of Typing Tutor is to demonstrate to people that the computer can adjust, remember and make elementary decisions. Another concept which is harder to transmit, is the limitation of the computer. People have to realize that the computer is not smart - does not think. Alot of mystique and some myths have to be debunked.

It is very important that the hardware be easy for the teacher to use. I usually recommend either cartridge systems or disk systems to reduce the hardware frustration for the teacher. Cassette tapes are slow to load, time consumming to backup and catalog, and frequently difficult to load. When I set up a diskette for a teacher, I always make sure it will run automatically and I limit the choices of programs to only a few. I have found that if I can give the teacher the impression that the computer can do only a few things, they are more comfortable with the machine in the beginning. Later, when they are comfortable with it, I give them a diskette with a variety of thirty programs on it.

The computer is really not a simple machine to run as any new teacher will freely admit. The first problem we face is the new terminology. Couple this with sequences or steps which are rarely spelled out in the operating manuals, and the fact that each program has its unique commands and codes, you have a bewildering maze of technological confusion. I find it useful to place labels on the different parts of the computer. This makes learning the part easier and helps when a teacher is reading the manuals. Anyone who brings a computer into a school had better be prepared to set it up to run with the least frustration.

For the teacher who is struggling with a computer on their own, I can only recommend that you try first to get help from the dealer, and failing that, try to find a users group for your machine. Usually someone in these groups will help the novice user. You can find out about them either through your dealer or through the many magazines on the market. I have known a couple teachers who just hung out at the computer store on a Saturday until some kind soul took them under their wing.

Finding the time to play with the computer is a personal problem faced by every teacher. I recommend taking the computer home for the weekend as I, myself, have seldom found any time while at school to play with the machines. In your own home you will be more relaxed and may have a more receptive spirit. Before you take the computer apart have someone show you how to take it apart and how to carry it in a car. I have had several tragic experiences with teachers, computers and cars. One teacher had to pay the school for $250.00 in repairs to a printer when it flew off the front seat of his car during a sudden stop. It also helps to take some masking tape and put matching numbers on all of the cables.

Purpose:
The computer is a multipurpose tool. It can be converted to different uses through the software one purchases for it. This software is nothing more than instructions to the computer to perform specific tasks. The software you select for your computer will determine the tasks you want the computer to perform. For this reason it is important that you sit down and decide what you are going to do with the computer. I usually define ten types of computer activity. They are:

1. Games - from simple to arcade
2. Computer Literacy
3. Drill + Practice
4. Computer Managed Instruction
5. * Simulations
6. * Word Processing
7. Data processing
8. Programming - Basic-Logo
9. Programming - High Lang.
10. Music + Art

I have added a star to the two areas which I feel offer the most immediate and dramatic potential for the computer.

Selecting a purpose for using the computer gives us a direction to aim for. I can't tell you how many times I have discovered that a school or teacher had no overall plan or purpose for using the computer. One startled teacher looked at me and stated that she was using a computer because all the other schools were using a computer.

From an simplistic perspective there are two general approaches to using the computer. You can either have the computer process the kids or you can teach the kids to use the computer as a tool. A good approach is a balance of the two. Frequently, I find that schools and teachers fall into the trap of only relying on software. Many schools start with computer literacy and then refine the purpose to a combination of programming and drill and practice. The teacher should answer some questions first. Will using the computer add some new dimension to what I am teaching? Is this the best tool for the job? Don't jump on the computer bandwagon until you have answered those questions - even if you are being pressured from above. There are some jobs the computer can't do.

As far as methods of using a computer goes, I have seen many different approaches. First there is the use of the computer as a demonstration tool where only the teacher uses the computer and the class sits and observes. Next is the limited class participation where a select group presses the keys, and the rest just watch. Then we have the team or group approach where two or more groups are competing against one another through a program such as Pork Barrel, The Discovery Series (Energy Search, Geography Search, etc), Snooper Troops, or Hamurabi. Another extreme is using the computer with only the gifted or learning disabled student one at a time during the normal class. Or there is one computer for each kid and all the kids working on the same problem or working on individual projects. Each teacher has to access their particular classroom and students decide on their own method.

Access:
Scheduling is one headache I have never overcome. I have tried the Use-it-Or-Lose-It rule

where each child has a scheduled time on the computer. Another time I tried only signing up those kids who elected to work with the computer. At one school the classroom teacher sent students to the computer lab with specific tasks to perform such as twenty minutes with a particular drill and practice program. In a class of thirty students and one computer, you are going to have to solve this dilemma. Don't be too rigid for in experimenting you will find a solution.

Software:
Selecting software is a problem in itself. Much of the software on the market today is poor. I try to select programs which allow two people to play against each other while the computer keeps track of scoring. I met one teacher who had only four commercial business programs which she used with her second grade class. She took a mailing list program and had it modified by a student to change the labels. Her children entered their name, whether they had a pet, sister, brother, the foods they liked. Then using the search function she had the computer print out all the kids with a pet and a sister. She used a word processor to draw pictures. Creativity will go a long way.

Getting started with a computer will not be an easy task, but it can be a rewarding one. There are some tasks the computer can do better than the human, but there are still many tasks only the teacher can do well.

SIMPLIFIED TUTORIAL PROGRAMMING FOR INTERACTIVE CAI MATERIALS

D. L. Jelden
Educational Technology
University of Northern Colorado
Greeley, Colorado, 80639

ABSTRACT

This paper is a continuation of an article written in the spring 1981 issue of the AEDS Journal which describes the microcomputer version of an educational model and the simplified dot commands which control interactive tutorial instruction. This presentation will show how the script generation is accomplished and how the utilization of the dot commands for the lesson can simplify programming problems, reduce computer memory capacity and manipulate students on unique pathways through the CAI lesson for the most effective and efficient learning experience in any subject field (Jelden, 1981a, 1981b).

INTRODUCTION

With the proliferation of microcomputers and the incompatibility of operating systems which control them, a simplified, compatible system of generating programs for tutorial CAI was needed. In examining the many types of computers, input/output devices and operating systems, one system in particular surfaced as the predominant one for microcomputers--CP/M and MP/M from Digital Research Corporation. In order to overcome many of the inter-system compatibility problems, a script generation model and dot command structure was developed and adopted to a CP/M-MP/M format on a microcomputer.

Although many restrictions still exist for micro applications in tutorial instruction, the existence of a structured teaching model which can easily be followed by the classroom teacher and which simplifies programming of material will provide for expanded use of CAI materials in the existing classrooms of today's schools (Jelden & Brown, 1982).

THE EDUCATIONAL MODEL

Any good teaching/learning operation is only as effective as the interaction process between the teacher and student. The teacher's responsibility is one of assessing the individual characteristics of the students. It involves getting to know the background of each student in terms of learning strengths and weaknesses, their likes and dislikes, their past experiences and success or failure in formal learning previously accomplished. In short, it is the obligation of the teacher to know well the students they attempt to teach. This process can be supplemented by external devices such as aptitude batteries and scales.

The other primary task of the teacher is to identify, order and prepare the content of the instruction in a manner which is acceptable to the student and which meets the objectives of the teacher. The anlaysis of content materials in line with the best we know in educational theory and practice can simplify the preparation of CAI materials. Establishing a learning sequence from a logical/psychological order will provide for greater educational efficiency of the tutorial lesson.

There are a wide variety of teaching/learning models in use today (U.S. Continental Army Command, TR-71-2, TR-72-1). The utilization of discovery learning at the lower elementary level might prove to be more personally rewarding to the student. It has been shown, however, that as students mature and move into the junior/senior high programs, a more structured format used in teaching the basics essential to high levels of competence is more effective. Students expect, and rightly so, the teacher to tell them what they are expected to learn. This attitude fostered the extensive use of performance or criterian objectives in educational programs over the last 15 years. The existence of this attitude from the students' point of view makes the utilization of the computer a natural support system for the educational environment. Now, the students can ask the computer what they are expected to learn and do it at a time pertinent to student needs, wishes and interests. The computer does provide a truly individualized, self-pacing educational experience for the student (Jelden & Brown, 1982).

CAI FLOW CHART

In order to better understand the nature of a lesson or program presentation, you might think of a presentation which includes three basic elements: (a) a pretest, (b) a lesson sequence and (c) a post-test (see Figure 1). The preconditioning, conditioning and prover unit sequence would be repeated for each new sub-element or major element of information found in the lesson. This system provides an opportunity for the learner to not only reach the objectives of the lesson, but will teach all the other enabling objectives required to achieve it as well. It is this flexibility, managed by the program system that makes CAI so effective. To simplify the explanation, the chart has been designed to show the alternatives for two extremes of aptitude, high and low. The arrows indicate possible paths depending upon student responses or system controls programmed by the instructor.

DOT COMMANDS

To implement a usable programming language for CAI on a microcomputer, a management program was written called ISSCAI. The heart of the CAI "coursewriter" system is the use of dot commands which are placed at strategic points within the scripts. A collection of dot commands available follows. These commands are placed at the beginning of a line, within the text of a lesson. If the first character of any line begins with a dot or period, then the line is considered to be a command line and it it scanned for the command. Each command is an abbreviation of the function it generally performs. In this way, it is helpful in reminding the programmer what the command does (Shaffstall, 1980).

Dot Command	Command Name
.AN	ANswer (W=wrong, C=correct)
.BL	BLind
.CC	Condition Chain
.CG	Clear Grade
.CH	CHain to new file
.CO	COmment (student)
.CS	Clear Screen
.EN	ENroll lesson
.EX	EXit lesson
.GN	Get Name (student)
.GO	Grade Off
.LB	Link Back
.LE	Lesson Element
.LF	Link Forward
.LK	LocK lesson
.OL	Override Lesson
.PA	PAuse
.PC	Pause Clear
.PS	Precondition Sequence
.PT	PreTest/Post Test
.PU	Prover Unit
.PW	PassWord
.QU	QUestion
.RC	ReCycle
.RE	Review Element

.SG	Show Grade
.TG	Test Grade
.TT	Terminal Type
.UA	Unknown Answer

The sample lesson element below is programmed for you with the dot commands so you may see what the script and the dot commands will look like in a programmed lesson. The lesson shown is not complete.

Sample Program

```
.CS
.PS
TO UNDERSTAND POWER WE SHOULD FIRST
BECOME FAMILIAR WITH THE TERM WORK.
WORK IS PERFORMED ANY TIME AN OBJECT
IS MOVED BY THE APPLICATION OF FORCE.
.PC
.CC
IF YOUR CAR WERE TO RUN OUT OF GAS,
YOU WOULD HAVE TO PUSH IT TO THE
SIDE OF THE ROAD.  PUSHING THE CAR
WOULD BE A FORM OF ---?---.
     A.  WORK
     B.  FORCE
     C.  ENERGY
CHOOSE THE CORRECT ANSWER AND TYPE
A, B OR C.
.QU 3
B
C
A
.UA
TRY THAT ONE AGAIN @.
.LB
.ANW
NO.  THE FORCE IS WHAT YOU APPLY TO
THE CAR  WHEN YOU PUT YOUR SHOULDER
AGAINST IT AN PUSH.
.LB
.ANW
WRONG.  THE ENERGY IS YOUR OWN
ABILITY TO PUSH THE CAR.
.LB
.ANC
WORK.  RIGHT, AND THE BIGGER THE
CAR THE MORE WORK YOU MUST DO.
.PC
.PU
WHEN WE CAUSE ELECTRONS TO MOVE
THROUGH A CIRCUIT BY APPLYING A
VOLTAGE, ---?--- IS BEING DONE.
    A.  WORK
    B.  FORCE
    C.  ENERGY
CHOOSE THE CORRECT ANSWER AND TYPE
A, B OR C.
.QU 3
B
C
A
.UA
OK, YOU NEED SOME HELP.  I'LL SEE
WHAT I CAN DO.  "#"
.LB
.ANW
.LB
.ANW
COME ON.  WE'VE ALREADY SAID THAT
FORCE IS A PRESSURE APPLIED TO AN
OBJECT TO MAKE IT MOVE.
.LB
.ANW
NO, NO.  ONCE AGAIN, THE ENRGY WOULD
BE THE ABILITY OF THE VOLTAGE TO
CAUSE MOVEMENT OF ELECTRONS.
.LB
.ANC
WORK.  CORRECT.  ANY TIME ANY OBJECT
IS MOVED BY FORCE WORK IS PERFORMED.
```

Note that with the use of a support statement to a correct answer, the learning is reinforced and provides for greater retention of the concept. The conditioning chain is designed to give the student a chance to make a connection between the information which they probably now possess and interconnect it to the new concept under study.

To find out if this connection has been made, a "prover unit" example is provided which calls for the student to respond to the new concept presented--that example being the definition of work as it is used in electrical application.

Greater detail about the system is not possible in this brief presentation. Just be aware that all of the management and recordkeeping functions available within the system can be contained and operated in approximately 8K of CPU memory space in the microcomputer. With this system approximately one third of the programming space is saved when comparing the same program written with the

BASIC language. ISSCAI is also far superior to BASIC in the simplicity of programming scripts and teacher data entry.

SUMMARY

With the advent of the micrcomputer, more classroom and educational applications of it are likely in the future. The ability of the computer to present and manage data for educational purposes is great. The main factor of CAI, however, is the quality of the material put into the system. With a valid CAI model to follow, the pretest element, the lesson elements and the review element do an effective job of teaching. Students who learn quickly can skip the conditioning sequence and go directly to the prover units within the lessons. Those who need extra help can receive it without slowing down the more capable students. Those students who need some additional support for selected items can receive it also. The large degree of flexibility within the system, however, is still related to the teacher's ability to write clear, meaningful instructional lessons. The author must take into consideration and anticipate the entry of the wrong answers some students may make. The teacher's ability to analyze, construct and implement valid lessons which anticipate student learning problems becomes the key factor ensuring CAI effectiveness in the classroom. Extended applications of the microcomputer in the future for tutorial interactive instruction programs can be expected. Even time-share microcomputer systems currently on the market make the expansion of CAI tutorial instruction within reason, economically.

BIBLIOGRAPHY

Jelden, D.L. "A CAI 'Coursewriter' System for the Microcomputer." Association for Educational Data Systems Journal, 14:3, pp. 159-168, 1981. (b)

Jelden, D.L. "The Microcomputer as an Interactive Instruction System in the Classroom." Association for Educational Data Systems Journal, 14:4, pp. 208-217, 1981. (a)

Jelden, D.L. and Brown, B.R. "A Generalized Learner Controlled Education System." Journal of Educational Technology Systems, Vol. II (1), Baywood Publishing Co., Inc., pp. 3-21, 1982.

Shaffstall, G. ISSCAI Reference Manual, Version 2.1. Lakewood, Colorado: International Software Service, 13050 W. Cedar Drive, #15, 80228, 1980.

U.S. Continental Army Command. An instructional model for computer assisted instruction. Ft. Monmouth, New Jersey: U.S. Army Signal Center and School, Computer Assisted Instruction Project, May 1971. (Technical Report 71-2.)

U.S. Continental Army Command. A summative evaluation of computer assisted instruction in U.S. Army basic electronics training. Ft. Monmouth, New Jersey: U.S. Army Signal Center and School, Computer Assisted Instruction Project, May 1972. (Technical Report 72-1.)

CONTRIBUTOR

Dr. David L. Jelden is Professor of Educational Technology at the University of Northern Colorado in Greeley. Dr. Jelden earned his Ed.D. degree at the University of Missouri. He has been a junior-senior high school teacher, then a university professor for 25 years, primarily in the field of electronics and educational methodology. His interest during the last 10 years has been in applying the microcomputer as an adjunct instruction system in the classroom. (Address: Educational Technology, University of Northern Colorado, Greeley, Colorado 80639, phone (303) 351-2169.)

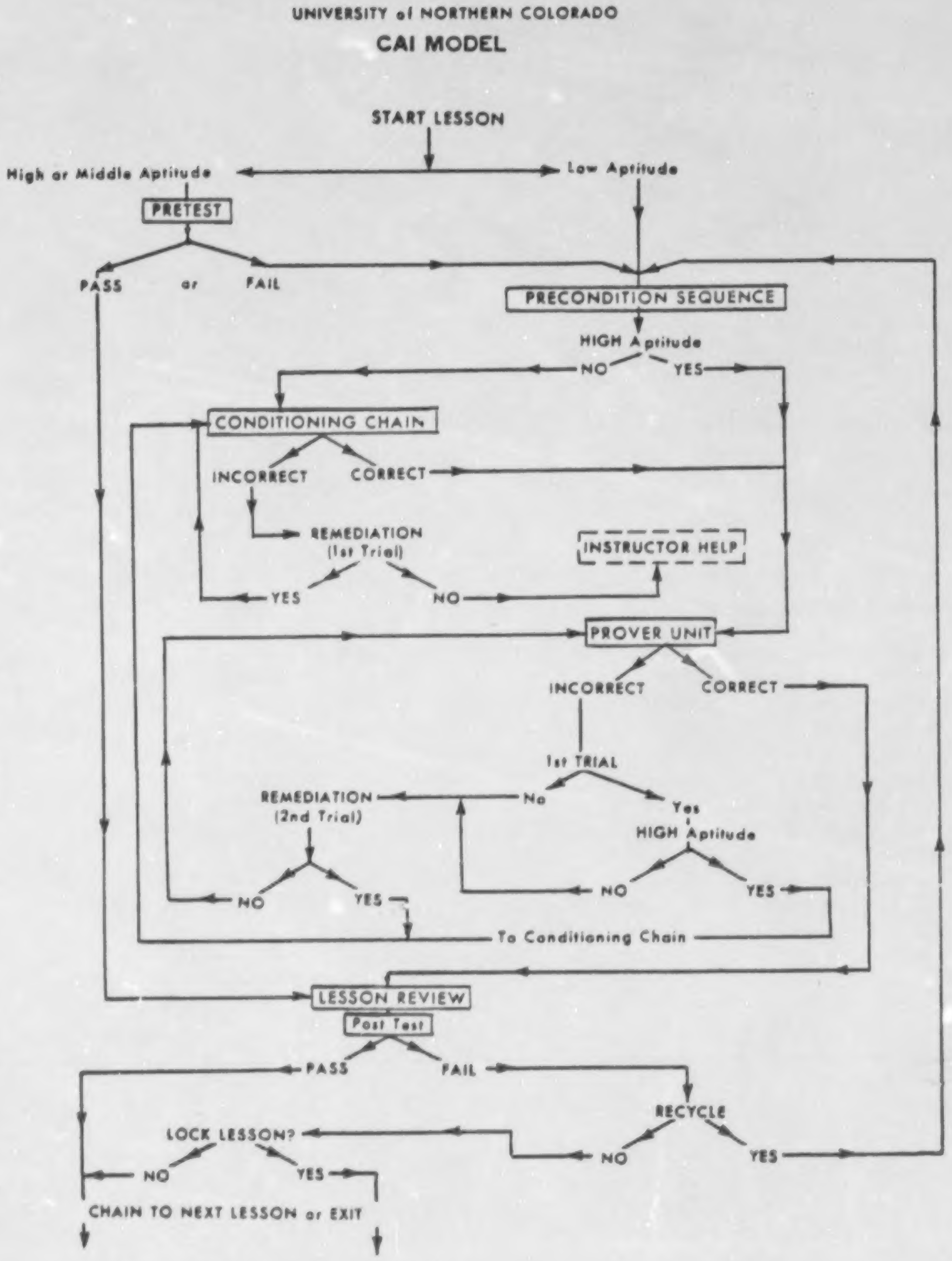

UNIVERSITY of NORTHERN COLORADO
CAI MODEL

Figure 1

Microcomputers at Home: Potential for Classroom Instructional Support

Dorothy H. Judd, Ed.D.
Department of Curriculum and Instruction
Northern Illinois University
DeKalb, IL, 60115

ABSTRACT
Three steps that teachers and administrators in K-12 schools can take to enhance the potential for home microcomputer support of classroom instruction are described.

Microcomputer use is sweeping across our society and filtering into the educational setting. Curriculum committees are or have been planning and developing processes for infusing computer literacy objectives into all subject areas or integrating a computer literacy unit into the existing curriculum. In the homes of America the microcomputer is finding a place in family life. The present sales battle waged by Atari, Commodore, Radio Shack, Texas Instrument, and Timex-Sinclair is bringing microcomputers within the range of a broader economic spectrum of American families. Moreover, the development and sale of keyboard units for the video "game" units widely sold in the past few years will mean further home computing power. The home or personal computer is no longer the exclusive province of the affluent.

The microcomputer in the home has important consequences for schools and schooling as we have known it. Almost any video game equipped with a keyboard, and even the smallest and least expensive microcomputers permit and encourage family members to engage in programming using the BASIC language. At this point in time, it is about the only thing the user can do with those models beyond using the unit to play computer games. For example, there are relatively few software programs available for educational needs that will run in the Magnovox Odessey unit. Complaints about the lack of software programs used to be heard about the Atari 800, Apple II+, Commodore Pet, Texas Instrument 99/4A, TRS-80, and other brands of microcomputers when they were finding their way into classrooms across the country. The change from little software of educational value to a tidal wave of significant educational software came while some were still wailing about its absence. We can expect the same phenomenon will take place in terms of home microcomputer use.

One of the disturbing phrases a teacher hears in the classroom, is from some bright students wildly waving their hands and reporting "We covered that last year!" Imagine the consternation that can occur when, as early as next year, several students interpose an objection to the topic for the day as something "they already have learned on their home computer." The structured curriculum of schools in the past, the teachers' tight control over the tools of their trade, such as textbooks, workbooks, and the like, was a guarantee that few parents would obtain these materials, let alone use them with their children. Just as widespread sale of home microcomputers has reached every corner of the country, widespread sale of microcomputer programs that are more than games will reach every corner of the country.

Sale of educational software programs for home use will be a reality as quickly as market research departments inform merchandisers of the market possibilities. Educators need to be alert for an opportunity to provide guidance to home computer users on the appropriateness of educational software. The parents who care about the computer literacy of their children will soon discover that "learning to program" in LOGO, PILOT, or BASIC is not all that the home microcomputer can do to supplement their children's classroom learning. If teachers do not share their knowledge about desirable microcomputer programs of educational value with the parents of school children, then parents are quite likely to make such decisions for themselves. These software purchase decisions will be based on advertising, sales pricing, and other non-educationally based criteria. If we want to have a positive role in shaping wise decisions by parents, we will need to prepare ourselves for the task.

Often the patterns of the past have discouraged the use of books and other instructional print materials that were school property. Thus, a bond that could have been developed between home and school instruction was overlooked. While there is still the opportunity, it seems a matter of prudence to begin to reinforce a school/home relationship in an electronic information age. It should be noted that home schooling has gained enough stature and respectability to be featured in a recent issue of the Phi Delta Kappan.[1] Also, the laws on school attendance have been construed to permit a large measure of home education in an increasing number of states.[2] For example, with microcomputers in the home, quality educational software programs, and the support of the local school board, the "neighborhood school" of tomorrow could be a cooperative venture between several families within a particular block.

The attitude that home computers compete with a teacher's authority role will not be easy for some of our fellow teachers and administrators to dismiss. However, teachers who accept the role of facilitator of learning as more useful to students and less stressful to teachers will not regard home computers as a competitive force. If teachers, students, and parents begin to think of themselves as partners in education, the introduction of the microcomputer into the classrooms of America, as well as the homes of our country will become a moment almost as important as the invention of the printing press.

A cooperative attitude on the part of teachers and administrators that will provide guidance and assistance to parents on the selection of appropriate instructional software for the home computer will assure better use of home learning as a supplement to school learning experiences. The teacher who wants to help parents in this way will be required to explore the availablity of various software programs. If this is not possible for the typical classroom teacher, the curriculum specialist for the school district may be able to provide direction for parents. On the other hand, if a microcomputer coordinator is on staff the appropriate software program information can be shared with parents provided this person is knowledgeable about curriculum matters.

Finally, the easiest challenge to cope with may be the idea of developing a school lending library of educational software. Obviously software programs must be available for teachers to use with one or more classes of students. However, in all likelihood, schools will find themselves with multiple copies of programs and the related documentation. Thus, this software could be checked out by parents of students for use in the home. If a software lending library function is not initiated by a school library, it seems likely that the public library in various communities may decide to undertake this function. As a last resort, parents in a neighborhood may band together in a formal or informal cooperative association for the purpose of providing educational software programs for youngsters in the area. Whether a school library, a public library, or a neighborhood cooperative association provides the software lending library service, the resources can begin with donations of software programs from parents. Since these donations have tax deductible implications for many citizens/parents, an effective appeal can be made to provide software for effective and individualized reinforcement of school learning experiences.

In summary, the constructive steps teachers and administrators in K-12 schools can take are:

1. eliminate any feeling of competition about the use of a home computer for the reinforcement of classroom learning;
2. cultivate a cooperative attitude toward assisting parents in selection of home computer software;

3. develop a school lending library of appropriate educational software and encourage software donations from parents to this library.

Now is the time for schools, teachers, and their administrators, to begin building a new climate for community support and to foster the acceptance of the role schools can play in the education of students in an information age. The problems to achieving this goal are challenging. However, the results of a growing partnership among parents, students, teachers, administrators, and microcomputers will be worth the effort.

---

[1] see John Holt, "Schools and Home Schoolers: A Fruitful Partnership," Phi Delta Kappan, 64:391-394, February, 1983; Diane Divoky, "The New Pioneers in the Home-Schooling Movement," Phi Delta Kappan, 64:395-398; Edward E. Gordon, "Home Tutoring Programs Gain Respectability," Phi Delta Kappan, 644:398-99.

[2] A survey by the National Committee for Citizens in Education, 1982, showed 38 states now allow home schooling, five more than did so in 1977; see review of John Holt, Teach Your Own, (New York: Delacorte Press, 1981), in Phi Delta Kappan, 64:441, February, 1983.

The Microcomputer: Analytical Tool for MBA Students

Robert C. Judd, Ph.D.
College of Business & Public Administration
Governors State University
Park Forest South, IL, 60466

ABSTRACT

The usefulness of microcomputers and business analytic programs to the MBA student are described. The availability and importance of electronic spreadsheet programs, decision analysis, production management, statistical analysis, and database programs are discussed.

The core of any MBA program comprises a series of courses that are the response by an individual college or school of business to the guidelines of the American Assembly of Collegiate Schools of Business (AACSB).(1) Typically the required MBA courses involve:

Managerial Accounting
Financial Analysis
Statistical Decision Making
Marketing Management
Production Management/Management Science
Organization Theory
Business Policies

You can readily see that the MBA degree, represents a generalist program of studies that stresses the development of analytic skills.

Since the microcomputer is a new tool in education at all levels, its use is only now beginning to take shape. The MBA program is no exception to this reality. Some of the software program resources that fit the MBA requirements of the AACSB will be described in this paper. Many MBA graduates hope to move into the corporate world where business microcomputer use is ahead of business school use at this time. The mid-level and senior executive in corporate life is using a microcomputer for its desk-top decision power. The most frequently used programs in business are:

* electronic spreadsheets;
* word processing programs;
* graphics creation programs;
* modem communication programs;
* database programs; and
* statistical analysis programs.

Leading in popularity is Visicalc,(2) an electronic spreadsheet program, that is available in versions for all of the major microcomputers. This microcomputer program provides for as many as 63 columns and 254 rows of data. Almost every executive has created a multi-columnar spreadsheet on a pad of paper in his or her earliest staff days. Experience with the problems of erasure and the re-adding of rows and columns is ample reason for their latter day enthusiasm for Visicalc and its look-a-likes.

Obviously students in MBA programs need to become very familiar with Visicalc or one of its counterpart programs. The special usefulness of this program is that a wide variety of mathematic functions can be performed on any value entered at a given row/column intersection. Thus, for example, it is possible to use Visicalc for almost any finance or accounting analysis function. Because Visicalc provides so many operating functions that are of importance in marketing, production and finance, a complete list of its functions may be useful. Arithmetic functions performed are the calculation of an absolute value, the exponent e, an integer, a natural log, a log to the base 10, and square root. Trigonometric functions performed are sine, cosine, tangent, arc sine, arc cosine, and arc tangent. Pi is the value, 3.115926536. Some functions of special usefulness in business are calculation of the arithmetic mean, the counting of all entries, the determination of the largest and the smallest value in a list of values, and net present value. In addition, Visicalc provides several logic functions involving True and False, equal to, not equal to, greater than, less than, as well as combinations of these tests. Further Visicalc provides such logic operators as Not, And, Or, If, Choose, and Lookup.

Visicalc has been described as a "one-man gang" of analysts. It is all of that and probably more. One of the most effective uses of the program is to interface it with a graphics program, a word processing program, or a database program and thus save having to key in data more than once. Values can be cross-walked from a database program to Visicalc, and then manipulated further. These same values can be cross-walked again to a graphics creating program to create such line graphs, bar charts, scatter diagrams, or pie charts as desired. Furthermore, the same values can be cross-walked to a word processing program and used to create an executive report. All of this can happen without running the risk that in retyping the figures, some error will slip in. Clearly a great deal of analytic power as well as executive convenience is found in using Visicalc.

Beyond the electronic spreadsheet program represented by Visicalc there are relatively few programs that will run in almost every brand of microcomputer. However, every major brand of desk-top system has database programs, word processing programs, graphics creating programs, telephone modem communication programs, and statistical analysis programs. Perhaps some further explanation of the terms used by some of the program types is in order. A database program is simply an electronic file cabinet with many drawers and many file folders in each drawer. In each file folder there are figuratively as many letters or documents as the user has planned for that file folder. The importance of a database system is that the names, addresses, and numeric values recorded can be recalled by using key words or symbols that make the retrieval of information much faster than any file clerk could perform the same task. Further, these numeric values and associated data can be used in Visicalc or similar programs that provide sophisticated manipulation of the information. Finally, the data can be used to create graphs and charts, or can be used to print out tables of information in an executive report. All of this is possible because of "one-time writing" of the data to a database and its subsequent linkability to other analytic or display programs.

Word processing and graphics creating programs are display programs that provide an executive with the power to create a letter quality report including graphs and charts from a printer that is activated by the word processing or graphics program. The telephone modem communications programs are becoming more important as electronic mail and other communications facilities become more widely available and used more often. The statistical analysis programs already available are as powerful in their way as Visicalc. Many of the best of these statistical analysis programs will equal the performance of SPSS(3) or similar programs. One of the best of these programs is StatPac(4) that runs in the TRS-80 desk-top system. Set forth below is a comparison chart in which StatPac is compared with SPSS. There are several unique features to StatPac that merit mention. First, any statistical analysis program written in BASIC, of particular use to a business user, can be annexed to the StatPac program. Thus, in the comparison chart, where a given function is not supported by StatPac, as for example is the case with discriminate analysis, the user can add the separate discriminate analysis program code lines to the StatPac system and thus enlarge the capabilities of StatPac.

Comparison Table

*=Superior Capability
A=Average Capability
0=Unavailable

Factor	SPSS	StatPac
Anova	*	*
Canonical Anal.	0	0
Cluster Anal.	*	0
Covariance	*	0
Cross Tabulation	*	*
Data Storage	*	*
Data Transform	*	*
Descrip. Stat.	A	*
Discrim. Anal.	*	0
Factor Anal.	*	0
Frequency Tables	*	*
Histograms	A	A
Large Database	*	*
Life Table Anal.	*	0
Mail Labels	0	*
Matrices	0	0
Meas. of Assoc.	*	0
Missing Data	*	*
Multiple Regres.	*	*
Non-linear Regr.	0	0
Non-parametric	*	0
Partial Correl.	*	0
Print Reports	A	*
Questionnaire	A	*
Regression Anal.	*	*
Residual Anal.	A	0
Scatter Plots	A	*
T and f Testing	*	*
Time Ser. Anal.	0	*

Second, StatPac, provides for either parallel or serial port output, and will accept input of 80 column card data via phone modem. Thus, data entry can be provided by a keypunch or CRT service bureau function and the data routed to the microcomputer by telephone lines. For those that do engage in data reduction and the creation of their own keyboard data input, one aspect of the StatPac program is a life-saver. This is the automatic backup of entries, so that no disaster can wipe out more than a very small part of data or commands entered into random memory of the TRS-80, because they have been automatically saved to a disk file. There are nine instances in the table above where the StatPac program does not have the capability to calculate the measure desired that SPSS does have. There are five instances where the StatPac program has superior capabilities to the SPSS program.

Specialized forms of microcomputer programs are available that can be important to the required courses listed earlier. It should be evident that the program software already described will take care of most of the course requirements with the possible exception of Production Management/ Management Science. Even in this field there are a number of programs available for MBA student use as well as business executive use. One of the best of these is the set of programs from the American Institute of Industrial Engineers (5) that provides such capabilities as calculating crew size analysis, assembly line balancing, line of balance analysis, and setup time sequencing. Another outstanding set of programs is called Decision System (6). In a single disk program that the author uses on the TRS-80, there are five choices:

- breakeven model;
- decision-theoretic model;
- lottery-insurance model;
- element-ordering model; and a
- game theoretic model.

The most useful involve a combination of using the element-ordering model to achieve some ranking of preferences and then to use the decision-theoretic model which provides for testing expected utility, the LaPlace criterion, the max-min criterion, the max-max criterion, the min-max criterion, and the Hurwicz index. These tools add to the analytic capabilities of any executive concerned with problems in production or management science areas, and they can be readily used by MBA students in finding answers to the analytic problems given in their textbooks.

Use of microcomputers in MBA programs is growing. This past year there have been more microcomputer papers and other presentations at regional and national meetings of professional societies in marketing, management, finance, and production than ever before. In an information age, this is as it should be. The encouraging thing about the new use of microcomputers by MBA candidates is that it should materially improve their capacity for analytic problem solving and decision making.

---

(1)Accreditation Council Policies, Proceedures, and Standards, St. Louis: American Assembly of Collegiate Schools of Business, 1982.

(2)Software Arts, Visicalc, Sunnyvale, CA: Visicorp., 1981.

(3)Norman H. Nie, et.al., Statistical Package for the Social Sciences, New York: McGraw-Hill Book Company, 1975.

(4)David Walonick, StatPac, Minneapolis: Walonick Associates, 1981.

(5)Gary E. Whitehouse, AIIE Micro-Software Production Control, Norcross, GA: American Institute of Industrial Engineers, Inc., 1981.

(6)Decision System, Lubbock, TX: Software Engineering Systems, Inc., 1980.

DISTRIBUTED PROCESSING AND COMPUTER NETWORKING AT BROWARD COUNTY - A STATUS REPORT

FRED H. KAUFFMAN, DIRECTOR
MANAGEMENT INFORMATION SYSTEMS
The School Board of Broward County, Florida
1300 West Las Olas Boulevard
Fort Lauderdale, Florida 33312

ABSTRACT

Broward County is proceeding smoothly with its schedule to fully establish a state of the art distributed processing network encompassing all 158 schools by March 1983. This network consisting of 170 Northern Telecom 585 minicomputers will, upon completion, enable all schools to manage student records and perform instructional management functions at their individual locations. As of February 1983, there are 103 Northern Telecom 585 minicomputers installed and fully operational at schools where they are being used for applications, such as student registration, attendance, generalized reporting, schedule changes, grade changes, and DISCOVER. At night this data is electronically transmitted to the central site to update the student data base. By March 1983 the secondary schools will be provided with remote job entry to the IBM 4341-2 mainframe computer to perform student scheduling and by April 1983 the district will be a pilot site for the Florida Information Resources Network.

This presentation will be of interest to all data processing personnel and school administrators interested in the why's and how's of installing a sophisticated distributed processing network.

BACKGROUND

The School Board of Broward County (SBBC) is currently the nation's tenth largest school district. It serves a population body of 124,000 students at 158 school locations. These locations break down as follows: 22 high schools, 28 middle, 97 elementary, 2 vocational and 9 exceptional centers. School based management has been in place for several years. Briefly, this means that decision making responsibility and authority as it relates to course offerings, staffing, and budgeting resides with the Principal. The curriculum is competency based and is called the Pupil Progression Plan (PPP). This means that a student must pass 85% of skills at grade level in order to be promoted to the next grade. Monitoring of these basic skills is reported by the teachers to be quite time consuming and tedious.

The central site computer complex currently consists of two IBM 4341-2 computers operating in shared DASD mode under DOS/VSE Rel. 3. There is approximately 3.6 billion bytes of direct access storage soon to be expanded to 6.6 billion. The central site scanning is performed on NCS 7018 and 7015 scanners.

Approximately 100 terminals are currently installed throughout the county offices, area offices, and the MIS department. Additionally as of 2/28/83 109 Northern Telecom 585 minicomputers (of which 103 are located in schools) have been installed in a distributed network arrangement.

Instructional computing is currently handled through the use of approximately 2,000 APPLE II Plus microcomputers. As a minimum each high school contains 30 microcomputers, each middle 20, and each elementary school 3.

Needs Assessment

Prior to the installation of the distributed network of minicomputers out to the schools student record transactions were submitted in batches to the computer center via interoffice mail. The secondary schools submitted their input via punched cards while the elementary schools submitted OMR scan sheets. All student related data processing was performed under defined production schedules consistent for all schools. In 1978 a criterion referenced instructional management/system called TRACER from CTB/McGraw/Hill was implemented to support the PPP by monitoring basic skills in reading, writing, and math. The input to TRACER consists of tests which are scanned and stored centrally. The output has proven quite useful to the teachers. However, the best turnaround that can be provided is 7 working days from when the teacher gives the test until the results are returned.

In summary, the problems of supporting a large decentralized school district such as Broward in centralized mode were significant:

- turnaround time was measured in days
- error correction process was cumbersome
- system outage affected all users
- predefined production schedules had to be consistent for all schools
- pinpointing data entry problems was difficult
- ad hoc reports had to be scheduled in advance

Four years ago the MIS department analyzed problems affecting its level of support such as these and performed a detailed needs assessment. The result was a set of goals defining a new long term direction. Briefly, the major goal was to define and develop an architecture which would ensure that student and instructional data be timely, accurate, and available in the appropriate format for the appropriate user audience.

Strategic Direction

To support this goal a five year long range plan was developed. The most significant aspect of the plan was the emphasis to be placed upon distributed processing as opposed to the large mainframe central site approach. This plan was presented to the School Board in March 1979 and approved. This approval marked the official beginning of distributed processing at SBBC.

The strategy chosen was to install a minicomputer in each school attached to a scanner and with sufficient terminals and printers to enable school based personnel to enter, update and retrieve data related to students. This network of minicomputers would be electronically linked to the central computer site so that that data from the schools could be electronically collected for management and external reporting. This implementation strategy was termed distributed processing.

Technical Approach

As proposed, the selected approach was for each of the 158 school locations to contain a minicomputer attached to a scanner. The configurations would range from 4 terminals and 2 printers in the high schools to 2 terminals and a printer in the elementary schools. The central site would contain 2 host minicomputers. These minicomputers would be used for file to file transfer of information with the remote network. Each day the schools would utilize their minicomputers to enter student data, maintain a local student data base, print out reports and perform instructional management. At night the host minicomputers would autocall each site and retrieve the day's student record transactions. This data would be passed to the 4341-2 mainframe via tape and used to update the central student data base. The edit/audit reports would then be passed to the minicomputers via tape. The host minicomputers would subsequently autocall the remote sites and transmit their reports. Then in the morning the schools would print out their reports. The file to file transmission process would also be used to transmit object code program updates and data files. The Houghton Mufflin instructional management system called ANSWER would be installed on each minicomputer. The schools would then be able to score tests and monitor the status of all program objectives for each student for the major curriculum areas.

The high schools, middle schools, and vocational centers would also be linked to the 4341-2 mainframe via a CICS bisync multi-drop line. This would enable these schools to utilize DISCOVER and to perform interactive inquiries against central data bases.

The functions this proposed network would be able to perform are summarized below:

- interactively create and retrieve student data resident at minicomputers
- interactively retrieve student data resident at central site computer
- scan and score tests
- print out reports locally both scheduled and ad hoc
- perform word processing
- transmit summary data to central site computer
- submit computer runs requiring heavy resources to central site computer for execution (i.e., computer scheduling)
- interactively use DISCOVER (career guidance system)
- enter/validate data to be submitted to central site computer

Current Status

Installation of the distributed processing network continues on schedule. As of February 28, 1983, there are 103 minicomputers installed and fully operational at school locations while the remaining 55 schools are being turned on at the rate of 12 per week. All installations are scheduled to be completed by March 31, 1983. Additionally, distributed minicomputers at Vocational Education, Division of Instruction, Budget Office, Warehouse, and Management Information Systems as well as the terminal network out to the area offices, Division of Instruction, and Department of Exceptional Education are fully operational.

All schools are currently using the capabilities of registering students, performing daily attendance, and producing a wide range of reports including student lists, rosters, and labels. The secondary schools in addition have the ability to inquire against the district's central student data base to retrieve student demographic data, schedules, and grades. Additionally they can retrieve enrollment summaries, class rolls, their master schedule, course dictionary, and teacher schedules and loads. All secondary schools now have access to DISCOVER, the on-line career guidance system. Prior to this time this capability was only available to the high schools.

The secondary schools additionally have been provided with the capability to enter and validate course dictionary, master schedule, student schedule, and student grade data. In addition, the high schools can enter graduation progress data. These capabilities have enabled the district to phase out all the keypunches and sorters in the schools and operate in a "cardless" environment. The resultant cost savings are about $130,000 per year.

In order to keep the district student data base in sychronization with the distributed data files at the minicomputers, each school is auto-called nightly from the central site computer and the day's activity is collected.

Upon completion of the updates to the central data base edit/audit reports are transmitted electronically to the schools. These reports are then available to the schools when they open in the morning.

Plans call for secondary schools to be provided with the capabilities to acutally perform their scheduling function directly from their minicomputers. They will be able to electronically submit scheduling runs to be processed on the district's mainframe and have the output printed on their respective minicomputers. This capability is in final testing and will be installed at the schools by March 31, 1983 in time for their 1983-84 schedules to be built.

The Vocational Educational Data on Occupational Students (VEDOS) system which enables adult/community school student registration and class rolls to be managed on the minicomputer has been extended to all 22 adult/community school sites and is also now fully operational.

The instructional management project is also moving forward. The contract for the software vendor to prepare Broward's functional specifications, implementation strategy, detail specifications, and project plan has been awarded by the Board and work is proceeding smoothly. The bid for optical mark readers for the schools has been issued and is scheduled to be opened March 3 1983. Current plans call for the instructional management system to be pilot tested in selected schools next fall.

Plans for Broward to be one of the two school districts for the FIRN (Florida Information Resource Network) pilot test are also proceeding according to schedule. By April of 1983 Broward will be able to transmit FTE data directly to Tallahassee via computer to computer transmission Additionally CRT terminals at Broward district offices and schools will be able to access software and data bases located at Tallahassee and the major state universities. Ultimately plans call for this statewide network to facilitate the transfer of information electronically between school districts, community colleges, state universities, Department of Education, legislature and the Governor's staff. It will be a leading edge for other states across the country.

Short Range Plan

The short term plan for the balance of calendar 1983 is as follows:

- make all remaining sites fully operational (March 1983)
- install remote job entry in secondary schools for scheduling (March 1983)
- award scanner bid (May 1983)
- install Version 2.2 in secondary schools to enable schools to maintain complete student record on minicomputer (August 1983)
- install scanners and test scoring application (September 1983)
- complete development of instructional management application (December 1983)
- include elementary schools in multi-point CICS network to provide on-line interactive access to central site data bases (December 1983)
- expand host data base to include additional data elements (December 1983)

Long Range Plan

Future plans call for the distributed network to include the following capabilities:

- access to Financial and Personnel/Payroll data bases
- district wide electronic mail
- entry of all school based transactions (i.e., requisitions, stock orders, personnel transactions, etc.)
- downloading of courseware from central library to APPLE II microcomputers interfaced to the minicomputers
- attachment of cash registers
- submission of ad hoc computer runs via RJE using EASYTRIEVE-like software tools.

Implementation Approach

In order to ensure successful installation of the network a careful, well articulated project approach was taken. The installation required careful coordination between Northern Telecom, Southern Bell, Paradyne, the Maintenance Department, and Management Information Systems. In order to accomplish this a formal project team was set up under a full time Management Information Systems project manager. Computerized project and installation schedules were developed and regular weekly meetings have been held with the Northern Telecom installation team. As a requirement of the bid award a full time Northern Telecom Systems Software Specialist is located on site until project completion. To ensure that all problems are logged, tracked, and reported, an on-line software package from IBM called A Network Management Program (ANMP) is being utilized. Thus all problems are immediately entered via on-line terminals at Management Information Systems, categorized according to problem area, and assigned to the appropriate individual. A variety of summary reports enables management to follow up on outstanding problems as well as pinpoint systemic problems.

The entire installation procedure was carefully developed so as to smooth out the changeover at the schools. A team of four School Liaison Specialists was set up within Management Information Systems. These individuals prior to this responsibility had extensive experience as data processing clerks in secondary schools. The function of this team is to inservice school based personnel, ensure a smooth installation process, and respond to problems in a timely fashion.

An inservice computer laboratory consisting of ten minicomputers, twenty terminals, and ten printers was set up in June 1982, and an inservice schedule developed. Two people from each school were brought in and provided with a full day of hands-on inservice as well as user documentation. They were also provided with a test cartridge to take back to their schools. This cartridge once loaded to disk was to be used by these schools to gain hands-on experience prior to be turned on live.

After a period of hands-on, a school attends a second inservice session where it is presented with a live cartridge consisting of its data from our central student data base. This cartridge is then loaded by the school and the school at that point is live.

Modifications to the software are performed on a release level basis. These updates can either be downloaded to the schools via electronic file to file transfer or a cartridge can be built and provided to each school. If the new release is a significant one (i.e., secondary schools data entry of schedule changes, grade changes, etc.) the schools are brought back to the computer lab for inservice.

Benefits

Benefits can be summarized as follows:

- Decisions can be based upon accurate, available, and timely data.
- Decisions of a broad scope will be based upon reliable information.
- Costly impediments to smooth information flow will be diminished, resulting in greater productivity.
- Teacher and system paperwork burden will be reduced.
- Curriculum can be more effectively monitored and altered to meet student needs. (Effective delivery techniques can be disseminated).
- Staff development needs can be more precisely identified.
- School-based managers will have the tools necessary to manage a school.

In summary, based upon comments from Principals, Assistant Administrators, and data processing secretaries Broward's distributed system has been a great success. Data is more timely, accurate, and available. Most importantly, personnel productivity has been increased. This state of the art project is receiving national attention from the educational community. To the best of our knowledge, it is the largest distributed processing network installed by any single local school district in the United States. Once again Broward County is becoming a leading edge in educational computing.

THE TWENTIETH CENTURY FRONTIER:
FROM BLACKBOARD & CHALK to MONITOR & KEYBOARD

Gary A. Kehrer and Nick A. Schepis
SCHOOL BOARD OF BROWARD COUNTY
1300 WEST LAS OLAS BLVD.
FT. LAUDERDALE, FL. 33312
305-765-6636,6766

ABSTRACT

This paper will delineate Broward County Schools efforts at implementing a districtwide computer education plan. The Broward School Distict is the nation's eighth largest. This plan encompasses the utilization of microcomputers in a lab environment; minicomputers in a distributed network; both of which are installed and operational in all district facilities. The in-service plan for the district's instructional and administrative staff will be addressed. Conceptual development, problem areas, and resolutions will be of interest to all district level and school-based personnel exploring the use of such technology.

BACKGROUND

The School Board of Broward County, Florida has invested over $11,000,000 in minicomputer and microcomputer systems. The minicomputer systems are installed in a distributed network linking ALL schools to the central office complex. The microcomputer systems are installed in a laboratory environment for instructional use in all schools. System configurations are as follows:

MINICOMPUTER SYSTEMS
Northern Telecom 585
$7,000,000 in equipment

MICROCOMPUTER SYSTEMS
Apple II plus 48K
$4,000,000 equipment

With hardware in place, we now could address the development of a comprehensive in-service plan for ALL district staff. This group includes central office personnel, school-based administrators, teachers, and clerical staff. The plan must take into account the varied levels of expertise of these users. The applications designed and software used must also be considered in the implementation of the plan.

BLACKBOARD & CHALK ERA

Educators have ALWAYS been slower to innovate than any other segment of our society. For whatever reasons, debatable as they may be, many educators still depend on traditional methods and pedagogies. Few attempt to give up this security through creative techniques. Grocery chains use computers & laser technology; fast food establishments use computers as point-of-sale terminals; banks utilize automated 24-hour tellers; the tourist trade industry uses remote terminal networks for scheduling; while school systems continue to operate with pencil and paper in hand. While these industries are using today's technology to attain their goals, schools are still inundated with performing tasks that are secondary to the prime directive: EDUCATE. Some of these tasks are: Attendance; Budgets; Local,state and federal reports; Inventory; Personnel files; Facility maintenance; Correspondence and reporting Student achievement.

Realizing the feasiblity of computerized assistance in these tasks, Broward County actively pursued change through the commitment from an innovative Superintendent and a School Board willing to pledge resources to this endeavor.

Yet, to affect change, a new relationship needs to be forged between knowledgeable Data Processing personnel and experienced educators. For years that relationship was one of mistrust. The DP department showed little concern about school-based applications. The school people lacked the understanding for the complexity and operation of a data processing shop. Before a sophisticated network of minicomputer systems was to be installed, the cooperation and trust needed to be established between educator and the data processing staff. This process began in Broward when committees were formed at the elementary, middle and high school level. These committees consisted of representatives from each school level and the DP staff. The educators began understanding the problems and considerations involved in computer systems. The DP'er began to have empathy for school-based management, average teacher salaries, grievances, & underenrollments. The turning point that finally solidified this relationship occurred in 1979...the year the microcomputer staked a claim in the twentieth century frontier.

MONITOR & KEYBOARD ERA

A major decision was made in Broward to go with the concept of Distributed Processing rather than many remote terminals connected to one mainframe. Distributed processing gives the schools the best of both worlds. The miniprocessor allows the school the ability to generate just about any report it needs from its own student files. These reports are generated right at the school and are based upon the needs of the individual school. One high school might need mailing labels of all of its seniors and an elementary school might require a mailing list for all second grade students in a particular teacher's class. The programs on the minicomputer are designed to give the user at the school the flexibility to generate a near endless lists of reports relevant to their school. Secondly, through the use of a modem and a lease line, the minicomputer is linked to the mainframe and gives the school the ability to look into its many DATA BASES. Any school can view the entire student data base, budget data and very soon the personnel and inservice data bases will be available. Students are using the career package "Discover" at all middle and high schools. This same minicomputer also has the power and storage for implementation of a full blown instructional management program. The Answer program, which has been a pilot for the past three years in Broward, will, by the start of the next school year, be piloted directly on the school's own minicomputer. The ANSWER system gives the school the ability to monitor a curriculum and to chart a student's progress in any of several curricula. Class reports, student reports and test results will all be handled at the school and controlled by the minicomputer. No longer will teachers have to wait for tests to be scored centrally or class reports to be generated by the central computer with a several day turnaround. If a teacher tests in the morning the results are in their hands the same day.

The minicomputer also gives the school the ability to perform remote job entry from the mainframe. The biggest use being student scheduling. The high and middle schools will collect and enter their data into a file in the mini. The school can edit this data up to a certain time window. This file is then transmitted over a lease line and the data is placed into a mainframe file. Each school then has the ability to access their file in the mainframe and submit it to run against the scheduling program. The school views its reports on their own terminal and decides whether or not to print. In most cases they view the results and change some data until the student conflicts are at a minimum. At this point they BUY their master schedule and student schedules are printed for each school.

Since 1979, a groundswell was occurring. More and more educators were using the microcomputer in their school. Inservice courses in micro applications were filled the same day advertisements came out. Principals, assistants, teachers, and the support staff could not get enough of this Third Wave technology. This all contributed to the raising of the level of computer literacy. When the staffs of school were inserviced in the use of the minicomputer system, a large majority of the people involved already had some hands on experience with micros. As a matter of side, pressing the ENTER key on one system was confused with pressing the RETURN key on another system. IPL and BOOTING a system entered the vocabulary of all principals. Storage mediums, whether floppy diskettes or winchester hard disks could now be discussed. Decisions about applications could now be implemented with both school and DP persons knowing the jargon. As long as the custodians of the mainframes are willing to share their expertise and knowledge and as long as the educators are willing to use this information processing tool to better help the education of "kids" the concept of BLACKBOARD and CHALK will make way for the new educational millenium of MONITOR and KEYBOARD.

Since the hardware was in place, the next major task was to provide literacy training for the eight thousand (8,000) teachers. Realizing that training such a large number of people in a short amount of time would pose an undesirable strain on county staff and facilities, the Broward County Computer

Education Committee recommended establishment of a group of teachers (one from each school) to be trained in the fundamentals of computer literacy. These teachers would then return to their school and in turn provide in-house instruction to their staff. This innovative approach to cadre training was met with enthusiastic response from principals. Those schools that already had a qualified person that could supervise the training at their school were exempt from having to send a teacher. Of 147 sites, 115 (78%) sent a teacher to participate in the county cadre. All training, whether for minicomputers or microcomputers, is given in a centralized laboratory setting with the majority of the time spent in extensive "hands-on" utilization of the equipment and their applications. The Broward County plan for in-service encompasses the interphasing of today's technology with the educational user community.

OBSERVATIONS

Over the past three years, we have had a chance to analyze our methods, staff and materials.The following summary should be of interest to all district and school-based personnel exploring the use of such technology.

1 A global understanding of what is to be accomplished by the computer system is necessary before the "nuts and bolts" can be taught to the participants.

2 The lines of communication between school personnel and DP departments must open up.

3 Administrators from High, Middle and Elementary schools should be on committees with the DP staff. These groups can discuss the interface of the Management Information Systems department and school applications design.

4 To place the users in a controlled environment, laboratories should be used for centralized inservice programs.

5 ALL personnel should be included in the inservice programs. County level, school based, instructional, administrative, and clerical persons should be allowed to participate.

6 Non-threatening vocabulary along with down-to-earth discussions of using the system, whether micro-, mini- or mainframe computer is encouraged. Instructional emphasis should be concerned with overcoming "computer fear/anxiety" syndrome.

7 School-based personnel should be hired as liaisons between the MIS department and the school level users.

8 All applications should be menu driven for user-friendly operation. This negates having a trained DP type at every school.

9 Computer literacy via the use of microcomputers can be a major stepping stone to a higher level of understanding of DP applications.

10 An effective inservice program excels by utilizing and supporting one and only one computer system.

GETTING STARTED - THE DEVELOPMENT OF AN ELEMENTARY COMPUTER MAGNET SCHOOL

Thomas G. Kerr, Ed.D.
Evans Computer Magnet School
Yeadon, Pa. 19050

This paper is a report of the development of an elementary computer magnet school designed to provide literacy and programming components as separate entities to four hundred (400) students in grades one through six (1-6).

The development of the Evans Computer Magnet School of the William Penn School District in Yeadon, Delaware County, Pennsylvania, has been simultaneously a most innovative, creative, challenging, and frustrating endeavor. This paper is part of an ongoing process to present the program as it presently operates in hope that it may serve as a conduit to other elementary projects or programs being created or already in existence. It is hoped that the problems and experiences encountered in the creation of the Evans Computer Magnet School will serve as a guide or reference point to others travelling down this interesting yet far from completed highway.

Program C.A.M.E. (Computer Awareness Magnet at Evans) was developed during the late spring of 2983 in an effort to resolve a court order to desegrate the school and to reduce the racial imbalance existing at Evans. When voluntary magnet options were reviewed it was made clear by the author that computers would be a most effective means of attracting students from the six communities comprising the William Penn School District. Each community holds a fierce pride in itself and views the neighborhood schools as effective and valuable.

The primary objectives of the program are to service all students (approximately 400) attending the magnet school and acquaint the students with the world of computers while infusing these newly acquired skills with curricular areas of language arts, math, science, and social studies. A major challenge to us as educators in the creation of this model is ensuring computer awareness skills while maintaining a solid instructional program in other curricular areas. The goal is to present this highly technical resource tool to students in a manner that allows them to understand and develop a sense of mastery of the microcomputers while building a bridge to the curricular areas.

The second major goal, that of reducing racial imbalance has been realized during the first year in that our minority population has been changed from 57% to 43% and we are optimistic about achieving a 35% cap within two school years. It is important to remember that the district and school achieved this change through a voluntary magnet program.

With this background, Program C.A.M.E. was created shortly after the March, 1982, Virginia Association of Educational Data Systems' Elementary Computer Seminar featuring the Minnesota Educational Computer Consortium (M.E.C.C.). From this conference specific procedures, techniques and ideas were provided by Ken Brumbaugh and Marge Kosel of M.E.C.C. and Randy Gull of Norfolk Public Schools. Seymour Paperts' Mindstorms (1980) provided the rationale and educational basis for our discussions and program, although L.O.G.O. is not the language selected for the magnet school.

During the summer of 1982, curriculum guides were written for both the literacy component as well as the skills component. These guides form the basis for the instructional program.

Each student receives an average of 105 minutes (grades 1-2) and 125 minutes (grades 3-6) per week with the computers. The program was created in phases with Phase I being the literacy (classroom) component and Phase II the programming skills (computer center) component. There are eleven categories in the literacy phase and six in the skills component.

Phase I - Computer Literacy

A. Everyday Use of Computers
B. Definition of Computers - Terminology
C. History of Computers - Famous People
D. Computers in Math
E. Computers in Science
F. Computers in Language Arts
G. Computers in Social Studies
H. Computers and the Consumer
I. Computers Impact on Society
J. Computer Careers
K. Flowcharting and Logic

Phase II - Computer Skills

A. Introduction to Computers - Hardware
B. Introduction to Keyboard - Typing
C. Introduction to Programming - PILOT
D. Flowcharting - Logic
E. Programming in the Curriculum Areas
F. Introduction to Graphics

One of the languages that was being marketed and was completed last spring for us to compare with our program goals was Atari's PILOT. This user friendly language provided the characteristics identified in our goals--ease of programming, graphics, and sound, all in a package ready when needed. The use of this language allows the program to focus on learning a language that can be applied to a specific curriculum. The process, therefore, became the product and the development of grade level objectives were created around this concept.

Each classroom teacher has the responsibility for teaching both components (literacy and skills) and developing and recording their specific lesson plans from our main curriculum guide. Grade level interpretation and record keeping of the objectives are essential for the success of the program. The use of regular classroom teachers to provide this instruction to our students has been, in the author's opinion, the single most significant factor in ensuring the success of the program.

Without the commitment, dedication, and interest of the staff this program would not have been possible. Each staff member was interviewed and selected to be part of the magnet program based, not on their computer knowledge and skills, but on their willingness to commit themselves to this new and exciting endeavor.

The staff has developed a greater sense of independence and self reliance since this program was introduced last spring. Their pride in the project is understandable and well founded. They have come to understand their own capacities for growth and the important role they play in this new educational field.

In-service training of the staff was supported by the William Penn School District during the spring, summer, and fall of 1982. Ongoing in-service work is planned around quarterly evaluations of the Computer Center.

It is through these evaluations that strengths and weaknesses are identified and new directions charted. The dialog between the staff is both stimulating and frustrating for there are many excellent questions raised yet so much uncertainty about their resolution. Once again, the process is the key to the product and in-service remains a key ingredient for the program.

The program makes use of parent aides and after school training sessions for selected students. Aides have been trained by the staff and assist in the computer center in a variety of jobs. The community evening adult school also makes use of the center and is presently teaching twenty-five community members about the computers. It is expected that aides, after school programs, and community programs will all expand next year.

The development of the program came as a result of a specific need and those working with Program C.A.M.E. believe we are moving forward to accomplish our identified goals. The program has been designed so that a conscious effort is made to support all students including those with identified learning disabilities. One has only to observe our students at work in the Computer Center to realize that the talents, skills, natural abilities and instincts are utilized immediately with this new and fascinating technology.

The program has received remarkably little attention from both magnet and neighborhood parents, despite significant media coverage, yet this is taken as a sign of encouragement and success. Recent surveys of four hundred fourteen parents, students and community members indicated an 83% success rate with the program and strong agreement to recommend the school. The program is designed to be infused with the curriculum and we

189

believe from these results that the program is successful and meeting the desired goals of instruction, desegration, and "computerization."

As the end of the school year is rapidly coming, major tasks for us remain in recruiting new students to the school for desegration purposes, readjusting the direction of the computer program, and continued in-service programs directed at specific, identified, needs. The search for new or better ways to operate the program are always being sought and this report is provided in hopes that a new supply of ideas to improve instructional computing at the elementary level will be forthcoming.

On behalf of the staff at Evans Computer Magnet School I welcome this forum and hope that new ideas, direction, concepts, goals or techniques may be identified and shared. This has been our real world experience--please share yours with us.

Microcomputer Selection

Dr. James E. King
Connolly Jr. High
2002 E. Concorda Dr.
Tempe, Arizona 85282

ABSTRACT

Since 1977, when the Radio Shack TRS-80 and the Commodore PET were introduced as complete micro units for less than $1000.00 the microcomputer industry has grown in leaps and bounds. Thus, it is no wonder that educators with little training in computers have found the plethora of microcomputer equipment and publications available a technological maze. In an effort to reduce the consternation that this can create, a doctoral study was conducted which produced a guide for the selection and acquisition of microcomputers by educational administrators.

The purpose of this presentation is to offer a review of that study along with selected excerpts which may prove useful to any educator faced with the task of selecting a microcomputer system for use in a school environment. The presentation will also address some of the myths that currently abound in microcomputer publications. In addition some attention will be given to current developments as well as developements that will be forthcoming in the microcomputer industry that may offer substantial benefits to educatiors.

Premises of the study

- Educational administrators may have to approve of disapprove requests for micros in their school.
- Educational administrators frequently lack computer awareness. In an informal survey at the state level in one state it was found: less than 10% of the sample understood computer terminology and potential applications of microcomputers and less than 5% of the sample had ever used a microcomputer.
- Few educational articles have been directed toward the school administrator. An ERIC search covering the years 1975 to 1982 yielded fewer than 5% of the articles on microcomputers that were addressed toward the school administrator.
- While several writers in the popular periodicals and in books on micros have stated the need for a computer needs assessment-very few specific procedures and techniques for conducting a needs assessment--vert few specific procedures and techniques for conducting a needs assesment were offered.
- Even with the rapid growth in the number of publications on microcomputers (150*) one could not go and obtain a single source and learn the essential elements prior to entering the micro selection process. One one profess or termed this a lack of standard reference materials.

- A guide containing a set of procedures and techniques for considering microcomputer selection could be developed for use by the educator with little or limited computer experience.

Procedures used in the study

- Review of the literature
- Develop an informal network of computer professionals, computer vendors, and educators who had used micros productively(but were not published)and utilize the information thus gained in the development process.
- Develop forms, instructions, and procedures used by beginners in computers for considering selection of a microcomputer system.
- Field test the forms, instructions, and procedures developed as a guide.
- Refine the guide with the feedback from the field testing.
- Have the modified guide critiqued by a panel of educational administrators who were not experienced in the use of computers.
- Produce the final version of the guide based upon the critique and field tests.

FINDINGS AND OBSERVATIONS

- the guide could not contain instruments that were too time-consuming to use.
- an information section preceeding the procedures and techniques was found necessary to help increase the computer awareness of the users of the guide.
- it was necessary to provide some specific examples of applications prior to begining the use of the needs assessment form (Form 1 of this report)
- it was necessary to indicate the minimal requirement of equipment and programs necessary for example tasks; i.e. word processing. This was defined as a minimal microcomputer system.

The Guide

The final (3rd) edition of the guide (appendix F of the King study) is composed of six parts. Part I provides an instroduction to the guide. Part II provides information about computer terminology and six popular (by# of sales) microcomputers. A description is provided of the micros and the functions of computer components with drawings. In this part the hardware and software are described with a comparison chart of the six popular systems. Examples are also provided of potential administrative applications. Part III decribes sources of other information from the more than 150 publications which may be of particular use for educators. Part IV contains a description of nine procedures that are recommended to the educator investigating microcomputers for their school or activity. Part V provided a description of the five forms developed specifically for this guide with examples of possible use. Form 1, data collection, is described very specifically for the following reasons:

1. it is the primary tool for conducting an in-house computer needs assessment.
2. it was validated in the study via educators who

critiqued the guide

3. it can serve as a description of each activity to potential vendors of microcomputers
4. it is based upon the "bottom-up" (Sanders, 1980) concept of takeing individual activities one at a time instead of trying to develop an overall comprehensive system and
5. a collection of several activities on this form can be used to help improve continuing manual operations or as a reference set for converting the activities to a microcomputer system.

A copy of Form 1 is included as the last page of this report. The other four forms are described more generally and were developed primarily from the literature search and the recommendations of computer professionals. However, from 4, the evaluation summary of bids, was recommended as addition to the guide during the first field test of the guide. Part VI provides a summary and tips for the potential microcomputer consumer. The warning 'Caveat Emptor' or buyer beware cannot be made too strongly for the investigation of microcomputer systems by the educator. The guide concludes with copies of the five forms.

SUMMARY

In the final analysis of any tool, of which the micro is but one of many the educator may use, the decision of whrther the device can be used effectively or not can be made only after a review of the activities for which the tool may be used. The efficacy of a microcomputer system will thus be related to the care that is taken in the assessment of activities for which the potential user is responsible and then selecting the system which most closely meets the needs and constraints facing the user(in this case the educator). The study reported in this paper (King, 1982) was an effort to develop a guide which could be used by educators (particularly school administrators) who lacked computer training to conduct a structured selection and acquisition process within their school without hiring external consultants. It was contended in the study that utilization of the full guide with the information section would increase the level of computer awareness by such users. It was also found in the study that the data collection form could be an effective starting point for educational administrators who were beginning to consider purchasing a microcomputer system. How effective the total guide is remains to be determined by future users.

References: for a full list of the references utilized in the King study please refer to that document. Those listed below by the author for their comprehensiveness and ease of use.

Brown, C.W. *The Minicompter Simplified* New York: The Free Press 1980

Joiner, L.M., Vensel, C., Ross, J. D., & Silverstein, B. *Microcomputers in Education.* Florida: Learning Publications, 1982. (one of the best beginning books for any educator considering microcomputer use.)

King, J.E. *A Guide for the Selection and Acquisition of Microcompputers by Educational Administrators.* Doctoral dissertation, Indiana University, October 1982, Dissertation Abstracts 1983

Sanders, D. H., & Birkin, S.J. Computers and Management in a *Changing Society.* (3rd edition) New York: McGraw-Hill Book Company, 1980

FORM 1: DATA COLLECTION

(1) Name of Activity/Task ____________________

(2) Responsible for Activity ____________________

(3) Assisted by (position/s) __________, __________

(4) ___ACCOUNTING/BUDGETING
___STAFF PERSONNEL
___STUDENT SERVICES
___COMMUNICATIONS

(5) FREQUENCY (check reporting period)
___Yearly
___Monthly
___Semester
___Grading Period
___Bi-weekly
___Weekly
___Daily
___Ad Hoc

(6) Description of Activity (action taken, reports written, etc. Please attach samples to this form if possible.)

(7) Reports of Activity to ____________, ____________, ____________

(8) Data (information needed to complete the activity/task

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

(9) Information Sources (people) ____________, ____________, ____________

(10) Related Sources (files, other reports, etc.) ____________, ____________

(11) Current Method used to complete activity requirements

___Manual ___External Service ___Computer

(12) Costs for this Activity (if known) $________.___ ___Estimated
___Actual budget

(13) Problems with current method (check those which apply)

___Accuracy ___Completeness ___Relevancy ___Timeliness ___Conciseness
___Control ___Other (explain) ____________________

(14) Benefits which may be realized by using a microcomputer system for this activity (check those which you would predict by using a microcomputer)

___Better use of educational administrator's time
___$ savings by reducing clerical commitments
___Better control of information flow
___More readily accessible information
___Less redundancy in maintaining files
___Improvement of: ___Accuracy ___Timeliness ___Completeness ___Conciseness
___Relevance
___Other (explain) ____________________

STRATEGIC PLANNING OF SCHOOL DISTRICT SMALL COMPUTER SYSTEMS

Chester Kiser
Department of Educational Organization,
Administration and Policy
State University of New York at Buffalo
Amherst, NY 14260

ABSTRACT

Most school officials have never had to plan or operate computer systems. Now they must do so to satisfy the demands of an emergent information society. This paper suggests a basic planning framework for a comprehensive school district microcomputer system. Included are practical suggestions on using the planning model.

NEED

Many school administrators seem perplexed by the microcomputer revolution and confused on how to cope with its opportunities and threats. Some would agree with the school official heard to say, "I don't even know the right questions to ask."

That's understandable, for we now stand at the threshold of a new era: the information society. Just as the industrial revolution freed workers' muscles, so the information revolution promises to free our brains from tedious tasks. With an abundance of information on virtually every topic easily at hand through computers and telecommunications, we can expect growth in human creativity and productivity (Masuda, 1981).

The opportunities educators face are to prepare students for that new society, both as information workers and as information consumers, and to use the newer electronic technologies to improve administrative decision systems.

The threats educators face are the technological, economic and decision demands of the information revolution. If permitted to become overwhelming, those threats could preclude effective school district adaptation.

School officials clearly need practical guidelines for envisioning a grand design of necessary computer systems, organizing planning efforts and implementing the desired systems.

PLANNING METHODS

"Planning is the design of a desired future and of effective ways of bringing it about." (Ackoff, 1970: 1.)

The diagram in Exhibit 1 sketches the components of a basic planning model that could inform and order the major tasks of developing a comprehensive school district microcomputer system.

Exhibit 1

STRATEGIC PLANNING MODEL FOR SCHOOL COMPUTER SYSTEMS

SITUATION AUDIT

WRITTEN GOALS

ORGANIZATION

WORK PLAN

ALTERNATIVES ANALYSIS

[Tactical Planning]

PHASING-IN

REPLANNING

The focus here is on strategic planning rather than on tactical planning. To differentiate between them, a strategic plan specifies 1) a desired future state that is years away and 2) ingenious, broad pathways for managing to get there. A tactical plan on the other hand specifies 1) a desired future state that typically is one year away, 2) logical procedures to operationalize the plan and 3) budget allocations to fund the plan. Although these two planning levels sometimes overlap, the distinction is useful for segregating macro and micro concerns.

Let's comment briefly on each component in the suggested planning model.

SITUATION AUDIT. Your planning premises constitute a situation audit (Steiner, 1979: ch. 8).

Ask yourselves at the start of the planning project: What is the situation we face in planning a computer system? Four categories of information sources can help to provide the answers.

ONE is expectations of major outside interests, e.g. society and local residents.

TWO is expectations of major inside interests, i.e. among students and staff.

THREE is environmental opportunities and threats, as mentioned earlier, e.g. opportunities afforded by the computer and telecommunications states-of-the art, and threats such as antiquated state education department regulations of local district computing.

FOUR is evaluations of school system strengths and weaknesses, e.g. staff and fiscal.

FIVE is a data base of forecasts and information on past and present performance, e.g. computer industry forecasts of trends and client's expressed levels of satisfaction with computing services.

WRITTEN GOAL. Ask yourself and staff: What is our grand vision of the instructional and administrative computing systems we desire within the next 3 - 5 years? That time span is suggestive; decide what planning period makes sense for you, so long as it is a multiyear one. Think big in wholes and ideals instead of small in parts and constraints only. View your intended system holistically. Write out your conclusions as a concise, clear goal statement. A few sentences or paragraphs might suffice.

ORGANIZATION. Constitute a Computer System Planning Council of interested persons representing the units likely to be affected by the new system. Charge the Council to function initially as a task force to develop a computer system master plan and later as a continuing body to maintain evaluative oversight of the computer system and to annually update the master plan.

Assign to a staff member the collateral duties of Computer System Coordinator to facilitate planning and implementation of the district's computer system. Also consider appointing Computer Helpers, by functional areas, to serve site colleagues as resource persons.

WORK PLAN. A plan to plan is essential. The Computer System Planning Council early should develop a Work Plan that details project objectives, related activities, target dates and responsibility assignments. The Work Plan might well schedule the project using a Gantt Chart or a PERT chart.

ALTERNATIVES ANALYSIS. The central work of the Council will be deciding issues on system applications, software, hardware, people and economics.

I suggest that system applications be planned comprehensively and district wide to include CAI, CMI, CAO and CAA. Computer Assisted Instruction (CAI) provides students with "private" drill and practice, simulations, modeling tasks, tutorials and basic programming skills. Computer Managed Instruction (CMI) functions as a teacher's aid to help with tasks of instructional diagnosis, prescription and record keeping. Computer Assisted Office Work (CAO) enables clerical personnel to electronically file, retrieve, sort and merge information. They also can electronically tabulate data, and edit and print documents. Microcomputers can contribute to office economy and effectiveness. Computer Assisted Administration (CAA) enables school officials to perform financial modeling, attendance analysis, test scores analysis, interactive inquiries of a data base and accounting tasks. CAA is a decision support system of much promise.

On software issues, don't spend much time deciding whether to make your own or buy it. You will buy virtually all of it. It simply is not in the cards for most educators to acquire professional level programming skills and then spend the hundreds to thousands of hours necessary to develop a single sophisticated program, either instructional or administrative application.

Only after you know what applications and software you want to run need you worry about hardware selection. Avoid the trap of too many school officials who buy computers and then ask what they should do with them.

Realize that you must plan the comfortable accommodation of your school system's staff people to the unfamiliar computer system. Alleviate theirs fears by involving them and training them.

The final step in computer system selection--after sequentially deciding on what applications, what programs for the applications and what computers will run the software effectively--the final step is selecting the best values among vendors' programs and machines. (Don't put the cart before the horse.) Decisions on values are based on the lowest bids of responsible vendors guaranteeing to satisfy your bid specifications. Finally, do budget realistically for the computer system, assuming annual additions and including at least ten percent of hardware costs for annual maintenance expense.

[Tactical Planning]. I included this parenthetical item on tactical planning in this strategic planning model just to show its sequential place in the total planning scheme. A full paper could be devoted to that topic, suggesting specific policy and procedural formulations for system implementation.

PHASING-IN. Its unrealistic to implement an ideal computer system throughout a school district in a single year. Plan to phase-in components of the system according to the sequencing decisions recorded in the multiyear master plan. When switching an office or administrative system, such as payroll, from operation, always run both systems in parallel until satisfied that the computerized system is functioning correctly.

REPLANNING. Planning never ends. Its a cyclical, iterative process. Update your multiyear computer system plan annually. Add a future year as you drop the past year. As suggested earlier, have the Computer Planning Council stay in business for that purpose.

SOURCES CONSULTED

Ackoff, Russel L. A Concept of Corporate Planning. New York: Wiley Interscience, 1970.

Masuda, Yoneji. The Information Society. Tokyo: Institute for the Information Society, 1981.

Osborne, Adam with Steven Cook. Business System Buyer's Guide. Berkely, CA: Osborne/McGraw-Hill, 1981.

Steiner, George A. Strategic Planning. New York: The Free Press, 1979.

Thomas, James L. (ed.). Microcomputers in the Schools. Phoenix, AZ: ORYX Press, 1981.

INTERACTIVE LANGUAGE EXPERIENCES BETWEEN STUDENTS AND COMPUTERS

by Gloria Kuchinskas Ed.D.

In the future, we can expect the microcomputer to have the same effect on language, on how we use it, and on how we think with it, as the printing press had on language in the Middle Ages. Seymour Papert (1980) says the computer of the future will be like a pencil: under the complete control of the writer, following the writer's will, not dictating to the writer and reacting with the writer in ways that are not possible with preprogrammed microcomputer material. The computer will not program the student; the student will interact with the computer.

John Krutch (1981), and other programmers interested in artificial intelligence, use the term "natural language processing" to describe this potential for the student and the computer to interact.

At first this idea may seem far-fetched to educators interested in reading and language arts, but interactive language programs for the microcomputer follow sound principles of language learning.

We know that microcomputers are highly motivational for students due partly to students positive experiences with computer gaming and partly to the American love of technology which is placing computers, with automobiles, in a special niche. We also know students produce better and more language when placed in motivational situations.

We also accept that students learn to speak by modeling the speech they hear all around them, receiving negative and positive feedback on their initial efforts and modifying language production based on this feedback. We do not always provide the same modeling experiences with written language production. Interactive language programs can provide such modeling experiences.

The production of interactive language programs is not new in microcomputer programming. By 1970, Florine Way had already developed DOVACK for a mainframe at Florida State University. This program implements the language experience approach to teaching reading.

Lucas'program, BABY SITTER, (1980) responds to any keyboard input from a toddler with the word, "Boo !"; a classic example of developing the concept that language is for communication on a very early level.

GUESS MY ANIMAL (Orlove, 1980) learns the characteristics of animals from the student and participates in a guessing game.

LEWIS CARROLL (orlove, 1979) uses a syllogistic approach. The program can reach a conclusion from the writer's premises and will even offer one that Lewis Carroll might have written.

Major Hugh Burns, at the Air Force Academy (Shostak, 1982) uses Aristotleian rhetorical logic to helps students think about what they will write in the prewriting stage of composition.

A number of poetry programs such as HAIKU (Krutch, 1981) or POET (Prentice, 1981) use random selection of words from parts-of-speech dictionaries to produce "poetry" that can be critically reviewed and rewrittten by students.

John Henry Martin (1981) has developed an interactive reading/writing program that purports to improve beginning reading instruction and that is currently being piloted by IBM.

Of course few of these programs reach the levels that Edmund Skelling, Florida's Poet Laureate, reaches with his marriage of color and poetry in Florida schools (Blair, 1982).

However it is not necessary to write a long, involved program to have a computer interact. The simple program that follows this article illustrates interactive language principles. This program calls the student by name, provides language feedback, learns from the student, enlists student support in the composing process, makes provision for student editing and revision and provides a hard copy for sharing and saving.

Such exercises should help students understand how language is generated in written form. After using such programs, students should accept the fact that they are in control and they are writers !

Bibliography:

Blair,M. The Best Will Recognize The Worth Immediately. ELECTRONIC EDUCATION, 1982.

Krutch,J. EXPERIMENTS IN ARTIFICIAL INTELLIGENCE FOR SMALL COMPUTERS. Howard W. Sams, Inc. 1981.

Kuchinskas,G. Developing Interactive Language Experiences For The Computer. THE COMPUTING TEACHER, Sept.1982.

Lucas,J.P. Programs For Small Children. CREATIVE COMPUTING, March, 1980.

Orlove,M. Guess My Animal. CREATIVE COMPUTING, Aug.1980.

Orlove,M.J. Lewis Carroll. CREATIVE COMPUTING, April, 1979.

Papert,S. MINDSTORMS: CHILDREN, COMPUTERS AND POWERFUL IDEAS. Basic Books, 1980.

Prentice,L.R. POET: A Program For Word People. CLASSROOM COMPUTER NEWS, 1981.

Shostak,R. Computers And Teaching English: Bits 'N' Pieces, THE COMPUTING TEACHER, 1982.

Way,F.L. The Language Experience Approach In Teaching Reading - Computerized. COMPUTERS AND AUTOMATION, Sept.1970.

----- A Conversation With John Henry Martin. EDUCATIONAL LEADERSHIP, Oct.1981.

* * *

```
10 ' This program motivates students to write stories.
20 '
30 CLEAR 2000 :CLS :PRINT @336, STRING$(33,"*")
40 PRINT TAB(18) "W R I T I N G  A  S T O R Y"
50 PRINT TAB(22)"by GLORIA KUCHINSKAS" :PRINT TAB(25)"copyright 1982"
60 PRINT TAB(16) STRING$(33,"*") :FOR I=1 TO 1000 :NEXT I
70 '
80 CLS :PRINT @330, "What is your first name "; :INPUT N$
90 CLS :PRINT @343, "Hello, ";N$;" !!"
100 PRINT TAB(15) "Use upper and lower case." :GOSUB 690
110 '
120 CLS :GOSUB 720  :PRINT TAB(23)"THE INDIAN" : PRINT
130 PRINT TAB(5) "The Indian stalked quietly through the dense forest."
140 PRINT TAB(5) "Suddenly he ..." :GOSUB 690
```

```
150 '
160 CLS : PRINT TAB(23) "THE INDIAN" : PRINT
170 PRINT "The Indian stalked quietly through the dense forest."
180 PRINT "Suddenly he ..." : FOR I = 1 TO 500 : NEXT
190 PRINT : PRINT TAB(10) "What do you think happened next, ";N$;" ?"
200 PRINT TAB(3) "Type your idea here. Press ENTER when you are finished."
210 LINEINPUT A$ :FOR I=1 TO 500 :NEXT
220 '
230 CLS :PRINT TAB(23) "THE INDIAN" : PRINT
240 LA$="The Indian stalked quietly through the dense forest."
250 LB$="Suddenly he "
260 PRINT : PRINT LA$ : PRINT LB$ + A$ : FOR I = 1 TO 1000 : NEXT I
270 PRINT : PRINT "What happened next, ";N$;" ? ";
280 PRINT "Press ENTER when you are finished." :LINEINPUT B$ :FOR I=1 TO 1000
 :NEXT I
290 '
300 CLS :PRINT TAB(20)"THE INDIAN" : PRINT
310 PRINT LA$ : PRINT LB$ +A$ : PRINT B$ : FOR I = 1 TO 1000 : NEXT I
320 PRINT : PRINT "Finish the story, ";N$;".";
330 PRINT "Press ENTER when you are finished." :LINEINPUT C$ :FOR I-1 TO 1000
 :NEXT I
340 '
350 CLS :PRINT TAB(20)"THE INDIAN   by ";N$ : PRINT
360 PRINT LA$ :PRINT LB$+A$ :PRINT B$ :PRINT C$ :FOR I=1 TO 1000 :NEXT
370 PRINT :PRINT TAB(5)"Look carefully at your story. You can change the "
380 PRINT TAB(3) "spelling, the words or the punctuation if you want to."
390 PRINT TAB(5) "Do you want to change your story, ";N$;" (Y/N) ";
400 INPUT R$ : R$=LEFT$(R$,1) :IF R$="Y" OR R$="y" THEN 460  ELSE 410
410 IF R$="N" OR R$="n" THEN 430  ELSE 400
420 '
430 CLS :PRINT TAB(23) "THE INDIAN   by ";N$ :PRINT
440 PRINT LA$ :PRINT LB$+A$ :PRINT B$ :PRINT C$
450 PRINT :PRINT TAB(15) "I like your story very much, ";N$;"." :GOTO 560
460 PRINT "How many lines will your new version be "; : INPUT L
470 IF L>9 THEN PRINT "Don't type more than 9 lines." :GOTO 460  :ELSE 480
480 PRINT "Retype the whole story; not the title."
490 PRINT TAB(10) "Press ENTER after each line."
500 FOR I=1 TO L :LINEINPUT S$(I) :NEXT I :FOR I=1 TO 1000 :NEXTI
510 '
520 CLS :PRINT TAB(23) "THE INDIAN   by ";N$ : PRINT
530 FOR I=1 TO L :PRINT S$(I) :NEXT I :FOR I=1 TO 1000 :NEXT I
540 PRINT : PRINT TAB(15) "I like your story very much, ";N$;"."
550 '
560 PRINT TAB(12) "Do you want a printed copy (Y/N) ";
570 INPUT Z$ :Z$=LEFT$(Z$,1) :IF Z$="Y" OR Z$="y" THEN 820  ELSE 580
580 IF Z$="N" OR Z$="n" THEN 600  ELSE 570
590 '
600 PRINT TAB(12) "Do you want to try again (Y/N) ";
610 INPUT Y$ :Y$=LEFT$(Y$,1) :IF Y$="Y" OR Y$="y" THEN 120  ELSE 620
620 IF Y$="N" OR Y$="n" THEN 640  ELSE 610
630 '
640 PRINT TAB(7) "Do you want to copy your story on paper ";
650 INPUT W$ :W$=LEFT$(W$,1)
660 IF W$="Y" OR W$="y" THEN GOSUB 710  :GOTO 690  :ELSE 670
670 IF W$="N" OR W$="n" THEN 690  ELSE 650
```

```
680 '
690 PRINT TAB(25) "Goodby ";N$;" !" :END
700 '
710 PRINT @980, "Press ENTER to go on." ;
720 LINEINPUT M$ :M$=INKEY$ :IF M$=" " THEN 720  ELSE RETURN
730 '
740 CLS :I=11 :A$=CHR$(191) :E$=CHR$(188)    'graphic for Indian head
750 PRINT @ I, A$; : PRINT @ I+64, A$+A$; :  PRINT @ I+(2*64), A$+A$;
760 PRINT @I+(3*64), A$+A$; :PRINT @ I+(4*64), A$+A$+A$+A$+A$+A$+F$+A$+A$;
770 PRINT @ I+(5*64), A$+A$+A$+A$+A$+A$+A$+A$+A$+A$+A$;
780 PRINT @ I+(6*64), A$+A$+A$+A$+A$+A$+A$+A$+A$;
790 PRINT @ I+(7*64), A$+A$+A$+A$+A$+A$+A$+E$;
800 PRINT @ I+(8*64), A$+A$+A$+A$; :PRINT @I+(9*64), A$+A$+A$+A$; :RETURN
810 '
820 PRINT TAB(12)  "Is your printer ready (Y/N) ";N$;" ";
830 INPUT X$ :X$=LEFT$(X$,1) :IF X$="Y" OR X$="y" THEN 840  ELSE 880
840 LPRINT TAB(20) "THE INDIAN   by ";N$ :LPRINT
850 IF R$="Y" OR R$="y" THEN 870  ELSE 860
860 LPRINT LA$ :LPRINT LB$+A$ :LPRINT B$ :LPRINT C$ :GOTO 690
870 FOR I=1 TO L :LPRINT S$(I) :NEXT I :GOTO 600
880 IF X$="N" OR X$="n" THEN 830  ELSE 830
```

A LEARNER-BASED COMPUTER INSTRUCTIONAL SYSTEM: AN EXPERIMENT

Seong-Soo Lee
University of British Columbia
Vancouver, B.C., Canada V6T 1Z5

ABSTRACT

A small scale learner-based instructional system was designed, implemented and field-tested, using Apple II plus microcomputer systems. Essentially, the system was developed in a study involving the task of learning the 'law of equilibrium' in the Piagetian balance scale form. The design of the system included three component aspects of a teacher: (1) Evaluator, (2) Diagnostician, and (3) Tutor, to be run under the control of a Manager. The Tutorial component was based on task analyses and developed into three modules to guide learning of relevant concepts, various rules, and problem solving. The present study tested only the effectiveness of the Diagnostician and the relevance of a certain system component. The results of the field testing involving fifty-eight 7th grade students indicated the expected useful function of the Diagnostician.

INTRODUCTION

Brown (1977) predicted 'a new science of learning and of teaching' would be emerging from research into the use of computers. He maintained that it goes beyond traditional psychology in its pervasive use of the procedural model of the learning process, beyond traditional education in its employment of new learning environments, and beyond traditional computer science in its focus on personal computing. Researchers in the area demonstrated the use of the computer as a consultant, as an assistant, and as a coach. To date, the main idea of using the computer has advanced from a frame-based system, to an expert-based one and further to a learner-based system.

Two typical examples of the learner-based computer aided instructional system, BUGGY and WEST, were illustrated by Brown and Burton (1978) and Burton and Brown (1981), respectively. BUGGY was designed to allow for the diagnosis of underlying causes of errors in a student's basic arithmetic skills. It was capable of telling not only which questions were answered incorrectly, but also why these questions were incorrectly answered. No tutoring was attempted. WEST (How the West was Won) was designed to be a computer-based tutoring/coaching system which has two major components: Recognizer and Tutor. The former was to allow for the diagnosis of a student's shortcomings from observing his behaviour; the latter based on the range of explicit tutorial strategies to say the right thing at the right time for effective coaching. Burton and Brown (1981) conducted an experiment by comparing the coached version of WEST with the uncoached, and showed that the coached group displayed a considerably greater variety of move patterns of a more subtle nature. Accordingly, the coached group enjoyed the game more than the uncoached group.

In my view, we have a long way to go to catch up with the rapid advances in hardware and software tools now available for the conceptual development of intelligent computer-aided instruction (ICAI) systems. The design of these systems should take into consideration four challenges, as identified by Glaser and Lesgold (1982): 1) procedural use of knowledge - - cognitive strategies, 2) how declarative knowledge comes into existence - - learning with understanding, 3) automacity and processing capacity, and 4) perception and representation - - a fast-access pattern recognition or encoding system.

In a way, Brown and his associates have incorporated the first challenge into their design.

A pilot study I undertook represents a small scale attempt to meet the second category of challenge: How to implement the idea of learning with understanding into a computer-based tutorial system. Learning with understanding requires active use of one's existing and relevant knowledge. The lack of understanding leads to inefficient trial and error and singular lack of transfer when problems are couched in different surface contexts. If a teacher (man or machine) is aware of a student's current level of understanding, it can be manipulated toward a more complete and more accurate level or toward an expert level. This can be accomplished through the learner's active mental operations involving inductive and deductive inferences. Records of learning processes provide the learner with critical feedback about conflicts between current and expert levels of understanding. Such instruction is a primary challenge for computer-based instructional designers. It requires use of the interactive intelligent medium to go from a passive information and problem presentation mode of instruction to an actively driving one. The learner is induced to change his concepts in the face of new information by formulating and testing hypotheses along the way that lead to new conceptual understanding. This kind of meaningful learning can be expected to occur in long-term learning situation(s) (Norman, 1980).

METHOD

In essence, the pilot study involved a task of learning the 'law of equilibrium' in the Piagetian balance scale form. The learner is expected to learn the Law and apply it to the relevant problem solving situation. Learning and application were considered to serve as the criterion of 'understanding'. The design of the instructional system involved 3 components of a teacher's role: (1) Evaluator, (2) Diagnostician, (3) Tutor. These three components were under the control of the instructional Manager in charge of pacing, record keeping, marking and reporting.

The Evaluator was to administer the pre-test of 36 structured items designed to measure the learner's current level of understanding relevant concepts and rules and also the post-test designed to assess the learner's mastery level, thereby assessing the effectiveness of instructional transaction. The Diagnostician was to identify 'bugs' in a learner's concepts and rules on the basis of each learner's characteristic pattern of responses to 12 clinical test items, to explain why those bugs were caused and to suggest a prescriptive instructional learning module. The Tutor consisted of 3 modules developed on the basis of learning task and process analyses. Module #1 was designed to coach the discovery of two relevant attributes (i.e., weight and distance) and the compensation rule (Piagetian concept of reciprocity) via 3 alternative submodules. Module #2 was to guide the formulation of torque rules and the learning of syntax of torque rules again via 2 alternative submodules. Module #3 was to coach the learner's process of solving problems couched in different surface contexts (i.e., lever and weight) which require transfer from the first two modules, that is the application or utilization of hitherto acquired concepts and rules. These modules were presented as guided discovery learning tasks requiring an inductive mode of thinking followed by a deductive application.

The Manager and three Modules of the system were coded in Apple II BASIC, using colour graphic displays extensively on Apple II plus microcomputer. The purpose of the pilot field testing was to assess (1) the adaptive effectiveness of the Diagnostician and also (2) the relevance of Module #3 as a task transfer. To this end, three testing conditions were defined. The Diagnostic group was given a debugging explanation followed by an appropriate submodule prescribed. The Non-diagnostic group was given neither debugging nor prescription, but all 3 modules linearly. Control group was given nothing but the first 2 modules. A group of 31 grade 7 students in a suburban public school was randomly assigned to the 3 treatment conditions defined above, 10, 10, and 11, to Diagnostic, Non-diagnostic, and Control, respectively. This field testing was replicated with another group of 27 grade 7 students of a similar school. They were assigned to one of the three conditions, nine to each. Both experimental testings of the system were conducted in school computer room environment. Each learning session lasted for 20 minutes. Learning events during each session were self-paced, with students having one session a day consecutively.

RESULTS AND DISCUSSION

The primary criterion measure was accuracy of performance on pre- and post-test. Although other response records during instructional sessions, only the preliminary results of analysing gain scores from pre- to post-test are shown in Table 1.

TABLE I
MEAN GAIN SCORES OF THREE GROUPS

	Diagnostic	Non-diagnostic	Control
S#1 (N=31)	7.00(5.31)[a]5[b]	2.90(4.70)2	2.64(3.17)2
S#2 (N=27)	8.44(5.27)5	4.11(4.01)2	4.11(4.81)3

a. Numbers in parentheses are standard deviations.

b. Numbers stand for those who obtained 34 or more mastery level out of a maximum of 36.

(S=School)

The first contrast of Diagnostic with Non-diagnostic group showed that the former treatment brought about greater gains significantly, *F* (1,52)=7.49, *p*<.01. The results of the gain score analysis clearly indicates that the Diagnostician of the system has worked as intended. The second contrast of Non-diagnostic with Control group showed that there is no observable difference in gain. This suggests that Module #3 (using lever and weight) was not working as a relevant transfer task to the criterion test (using balance scale form). The problem solving situation couched as a 'far' transfer task in different surface contexts must have had an unknown number of factors that are not conducive to students' understanding of concepts and rules in the Piagetian balance scale form. The present experimental field test of the learner-based system clearly shows a great deal of promise as a basis of ICAI system. We have just scratched the surface of the project that requires first the development of a conceptual scheme based on sound learning and instructional theories in information processing framework. The system developed in the Applesoft BASIC environment, having been modified for technical efficiency and user friendliness requires a larger and more sophisticated programming environment.

REFERENCES

Brown, J.S. Computers in a Learning Society. Testimony for the House Science and Technology Subcommittee. October 13, 1977.

Brown, J.S. & Burton, R.R. Diagnostic models for procedural bugs in basic mathematic skills. Cognitive Science, 1978, 2, 155-192.

Burton, R.R. & Brown, J.S. An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (Eds) Intelligent Tutoring Systems. London: Academic Press, 1981, pp. 79-98.

Glaser, R. & Lesgold, A. Psychological

and pedagogical issues for computer technologies in education. A Keynote Address at CBE Research Conference. University of Delaware, June 3-4, 1982.

Norman, D.A. Twelve issues for cognitive science. Cognitive Science, 1980, 4, 1-32.

LAW AND ORDER ON THE MICRO FRONTIER:
A SHARED DISK SYSTEM FOR EDUCATIONAL MICROCOMPUTERS

By

Paul Lorton, Jr.
University of San Francisco
San Francisco, CA 94117

ABSTRACT

Using a shared disk network of microcomputers in a secondary school is the hardware solution to many of the problems encountered in trying to manage several independent microcomputers. However, the hardware does not come with easy software solutions. In fact, a shared disk network of microcomputers requires overcoming several very interesting problems before the promise is fulfilled.

For two years, most of the time spent in a comprehensive high school in San Francisco, we have been developing the software to fulfill the potential of shared hard disk system. The problems we have had to overcome and the programs we have implemented for this system are varied and will be discussed.

INTRODUCTION

Microcomputers are being used extensively to bring the promise of computer use for instruction to the student. Because they are relatively inexpensive, often costing about the same as a timesharing system terminal, they can be acquired simply on the basis of their attractiveness to one or two individuals in a school. Often single microcomputers appear in an individual classroom because one teacher made the decision and took the initiative to acquire it.

Those of us with experience in classes with instructional timesharing terminals can recall dimly the development of those resources and can vividly remember what they look like now. Timesharing systems can be made quite supportive of the naive user. Software systems have been developed for the timesharing environment which allow easy access to the full range of programs available on the system. Such access has been made a simple task for the user so that an individual does not need to become an expert with the computer system in order to use the resources available.

The core of a timesharing system is one central computer through which all computing transactions must pass. This centralizes support and control. It also allows for very elaborate programs to monitor and guide all activity on the system. Only those people and programs operate at the user stations which the central system allows to operate.

Microcomputer environments are unlike timesharing environments because the user stations are not timesharing terminals, they are real computers. Because the computing in a microcomputer system is in the user station, the central control and support is lost. Each microcomputer is a self contained computer and must replicate the whole computing environment. From the orderly, controlled world of an educational timesharing system, microcomputers have brought us to a world in which disorder and duplication are the rule and even encouraged. To share software with other teachers or even students using several microcomputers physical copies must be made. Since this process involves individual contact, the effort of others, even in the same building, often goes unnoticed and many creative individuals spend difficult hours developing exactly the same instructional program.

It is an exciting, frustrating environment. The purpose of this discussion is, however, to take some of each aspect away from the instructional microcomputer experience. By reporting on the experience of using a shared disk system at a comprehensive high school in San Francisco, we hope to indicate ways in which many microcomputers can be used in an environment which restores some of the best aspects of timesharing systems while not detracting from the strengths of microcomputers as classroom tools. The main points we will touch on include:

1) Managing a shared disk system through user developed software approaches
2) Menu-driven system and software access
3) Multiple access to shared software and data
4) Effective curriculum development in computer literacy for faculty and students, and
5) Procuring appropriate software which can be used on a shared disk

THE COMPUTER SYSTEM AT GALILEO HIGH SCHOOL

In order to serve students more effectively under Chapter 1 of the Education Consolidation Improvement Act (formerly Elementary and Secondary Education Act, Title I) and the Economic Impact Act/Limited English Proficiency program, Galileo High School, San Francisco, has acquired various computing resources.

The purpose of the two federal acts, in general, is to serve the special needs of schools who have a large number of students who are achieving below the mean on standardized tests of basic skills. In addition, the special needs of schools with large numbers of students with limited ability in English are to be met.

The computer system at Galileo has two main components - a Digital Equipment Corp 11/70 and an Apple II - Nestar shared disk system. Ultimately, the systems will be linked to provide a coordinated, communicating resource and the most comprehensive computing utility possible with the on-site hardware. At present, however, each component is evolving to perform the functions for which the particular component was acquired.

DEC 11/70

The DEC 11/70 is funded by three sources, school site ECIA Chapter 1 funds, central office ECIA funds and San Francisco Unified School District funds. The three sources of funding mean that the system has three separate areas to serve: instruction at the school site, evaluation for the central office and service to other district schools.

Apple II - Nestar Shared Disk System

The Apple Network was acquired with ECIA Chapter 1 funds and is used mainly to support instruction of ECIA eligible students. To accomplish the goal of providing effective instruction through microcomputers, a variety of projects are underway. These include:

Development of Network Software

Using considerable experience from a parallel system at the London Business School, a number of utilities have been implemented to make network use simple and controllable.

Instruction in Basic Programming for Teachers

A set of material has been developed and is constantly being expanded to help Galileo teachers become familiar with the potential and use of microcomputers as tools in direct support of their own instructional efforts. The material used in this instruction was developed with partial support from the Apple Educational Foundation.

Software Evaluation

Much software exists for the Apple II. However, using the available software for high school instruction involves more than simply acquiring it. There is a continual effort at evaluating the existing software in conjunction with the courseware development effort.

Courseware Development

A final aspect of the effort to develop a useful microcomputer based instructional system is the development of courseware appropriate to the needs of Galileo High School. There is a current effort, for example, to provide computer literacy for the ECIA eligible students. Other such efforts will develop out of an increasingly broad community of teachers who are acquainted with the potential of the microcomputer system.

The supervision of the development of the Apple - Nestar shared disk system is the responsibility of Robert Lew, a science department chairman, who has taken on the additional challenge of producing an effective instructional environment from more than twenty-five Apple microcomputers.

THE SHARED DISK ENVIRONMENT

A shared disk environment can solve one of the most difficult problems associated with having a lot of separate microcomputers in a school. That problem involves how to provide the programs for student's to use without having to have many copies of the same diskette. In addition, the problems of building an easily accessible library of useful programs can be handled with a shared disk system.

The shared disk environment allows many microcomputers to have access to the same electronic source of material with little trouble. That "little trouble," however, does present its own series of problems which must be solved in order to make effective use of the resource. The problems we have encountered include:

1) availability of software for the network,
2) ease of access to the available software, and
3) speed of response of the system

Software Availability

Many of the programs which teachers would like to use are designed to operate on single microcomputers. Moreover, to protect the program author's investment, many of them are not copiable. In order to use a program on the shared disk system it must be copied onto the system and, to solve the second area of difficulty, it must be modifiable to fit into the needs of the shared disk access system.

Ease of Access

As soon as any useful program is acquired, the problem of how to make it available to the using community arrives. For a sophisticated user, learning how to discover and access programs on the system is not a problem. But such access is neither easy for the naive user to learn nor is it desirable. Software which guides and controls access to the software on the system is needed. As a readily appreciated example, controling access to the many attractive arcade-type games available for the Apple is a constant struggle. An equally important problem in access is keeping the teachers informed of what is available for use by categories which make sense from a curriculum standpoint.

Speed of Response

The ordinary use of a shared disk system is in an environment where the access is asynchronous - that is, users make requests of the shared disk at random intervals, spending most of their time working with a program in their individual computer. The use of the system in an educational environment means that, at times, all the student stations will want some access at the same time. For example, when a classroom of microcomputers starts for the beginning of a period, the teacher may wish all the students to begin at the same point. An initial test of the system at Galileo pointed out how difficult this problem can be. When 18 of the Apples connected to the shared disk were started simultaneously, as would be the case with the start of a class, it took between five and ten minutes for all the Apples to be loaded with the desired programs.

These three problems arise out of having a shared disk system. They are small compared to the problems involved in coordinating many "stand-alone" microcomputers in a school environment, but they are significant especially in that their solution often involves sophisticated programming approaches in an environment where such sophistication is not easily achieved.

SOLUTIONS

The solutions we have developed are in three main areas which cover the kinds of problems we have encountered in using microcomputers in an instructional environment and in using a shared disk to coordinate those microcomputers.

System Utilities

The ordinary software provided with shared disk systems provides the tools for accessing the system in the most general way. Most educational uses require very specific accessing capabilities. That is, a teacher will want a student to use a specific program. To get to that program should be a very simple task and should not require the student to learn anything about the shared disk system itself. This is also true for teachers.

To meet this requirement, a Menu System was designed and implemented with supporting software which allows sub-menus to be easily added into the system. In addition, the programs in the system have been designed so that the user is always supported in making requests of the shared disk resource. Control is established by insuring that the user's access to the system is almost always under the control of a program that is part of the system. The only, limited, exception to this rule is the case where the user is writing programs on the microcomputer.

To meet the additional requirement in speed of response as well as access control, a reservation system was implemented as part of the menu system which, in addition to reserving any or all of the networked microcomputers for any time and day, will preload a teacher specified program in any microcomputer at a specified time. Thus, turning on the Apple before a class enters the room is enough to insure that each Apple has the correct program in place for the start of the class.

User Software

There are three problems with providing sufficient user software: copy-protection, evaluation, and support of local efforts. The last problem is taken care of by the teacher training effort mentioned below and by the News feature of the Menu System.

Evaluation of available software includes both assessing its quality and appropriateness, and categorizing it for inclusion in the Menu System. Because the software is available on the network and easily accessed through the Menu System, this continuing process is accomplished with much greater ease than if it could only be accomplished by physically circulating diskettes.

Copy protection makes some software unavailable for network use. Being aware of this, software is only acquired when arrangements can be made for using it on the shared disk. Software publishers are slow to adjust to the shared disk environment but they are beginning to have coherent policies for shared disk use of their material.

Teacher Training

Training teachers to use the system is the most important factor in how successful the microcomputers will be in satisfying the educational goals for which they were acquired. It is a constant process. Classes are held during preparation periods and after school. Last summer, classes were offered and the classes will be repeated this summer. The most significant outcome of the classes is not the expertise individual teachers develop but the community which develops among them as they work with the microcomputer material.

RETURNING THE FOCUS

The experience at Galileo High School in San Francisco represents a potential solution for all schools faced with how to effectively provide school-wide computer services with many microcomputers. Much of the current effort at Galileo High School is centered on implementing the overall plan for the computing resource and fulfilling the purposes for which the system was acquired. The solutions we have developed mean our main effort, developing an effective microcomputer based instructional system, can be accomplished with minimal peripheral concerns. We have returned some order to the instructional computing environment. Hopefully we are accomplishing this without reducing the excitement microcomputers generate through their classroom use.

THE DESIGN AND USE OF COMPUTER LEARNING GAMES

Charles Lund
St. Paul Public Schools

Abstract: The design of learning games focused on the Apple II microcomputer will be discussed along with suggestions for the effective use of such games in the elementary and junior high school classroom.

Electronic computing technology is rapidly evolving. In this presentation we will first take a quick look at the big picture of computers in education. Then we will briefly examine the many roles microcomputers can play in instruction. Next we will explore the role of using educational games in some depth. Finally, we will take a look at a few of our favorite games and how they are used in elementary and junior high school classroom.

Computers in Education

There are at least three major uses of computers in education. They are:

Administration
Research
Instruction

Within each of these broad categories of use are dozens of microcomputer uses that could be a topic for discussion. For example, under the administration category a few possible topics are: word processing, electronic mail, electronic file cabinet, applications, phone directories, student accounting, scheduling, resource directories, and payroll. In these difficult economic times we are all interested in making sure that an application area like payroll is done properly, but this paper will focus on instructional uses of the microcomputer.

Sample Microcomputer Uses/Instruction

In this paper the term "instructional computer uses" is used to include direct use of the computer by students, teachers, aides and volunteers. Some of these uses are described below. The number and variety of instructional computer applications is growing rapidly and so this list is by no means exhaustive. It should also be noted that each category of use is listed as though it is disjoint from the others. In actual practice many applications feature more than one role.

Student programming	- Students learn one or more computer programming languages.
Tutorial	- The computer provides direct instruction, and the student responds to questions posed by the computer.
Drill and practice	- The student interacts with the computer in short drills or in a more comprehensive program involving the computer in a management function.
Simulation	- From a variety of disciplines, experiences that are too broad, complex, inconvenient, or dangerous for actual exposure to students are portrayed on the computer.

Educational games - Concepts are learned or skills are practiced through lessons in game format.

Data Base Inquiry - The computer provides needed information for guidance, career planning or other purposes.

Test scoring and item analysis - The computer scores tests and analyzes the data.

Management systems - The computer handles data and provides the teacher with information that is helpful in the management of instruction.

Office automation - Word processing, electronic mail, electronic file cabinet, scheduling, resource directories, phone directories, student accounting, and payroll.

Definition of Educational Games

The main focus of this paper is on educational games. For the purpose of this paper, educational games are defined as those activities that go beyond mere entertainment and may combine several of the categories of computer use listed above. We also believe that it is important to make it clear that we are a multi-usage district with a broad set of computer uses. We do not believe that one category of use or any one language is "the" answer. In short, our approach is eclectic.

Why Use Microcomputer Games?

There are many reasons for using microcomputer games in the classroom. A brief list of objectives is provided below. Note that many of the justification statements that are listed are not new. Computer activities are selected because they fit into our instructional goals and objectives.

WHY MICROCOMPUTER GAMES?

- TO REINFORCE GOOD TEACHING THROUGH INSTANT FEEDBACK.
- TO BUILD DESIRABLE ATTITUDES TOWARDS ARITHMETIC/MATHEMATICS.
- TO MOTIVATE.
- TO PROVIDE FOR INDIVIDUAL DIFFERENCES BY NUTURING LOGICAL THINKING AND STRATEGY INFORMATION.
- TO PROVIDE 'RANDOM' SKILL BUILDING OR CONCEPT DEVELOPMENT PRACTICE IN A PLEASANT SETTING.
- TO PROVIDE SKILL MAINTENANCE.
- TO ADD ANOTHER TOOL TO THE TEACHER'S REPERTOIRE OF TECHNIQUES FOR PROVIDING VARIETY IN THE CLASSROOM.
- TO PROVIDE APPROPRIATE "HOMEWORK" FOR PARENT-CHILD ACTIVITY.

Despite the fact that many of our reasons for using microcomputer games are the same as using any game or activity in the classroom, the dynamic interaction, random capability and "halo" effect associated with computer use, do provide a significant advantage over other media.

Characteristics of "Good" Computer Games

The fact that there is a plethora of software now available on the market make it important to have some criteria on hand when examining game software for possible purchase. Listed below are some of the characteristics that St. Paul teachers have indicated are crucial to consider when making a selection.

CHARACTERISTICS OF "GOOD"
COMPUTER MATH GAMES

- MENU DRIVEN.
  (NO PROGRAMMING EXPERIENCE NECESSARY)
- CLASSROOM TESTED.
- EACH DISKETTE SHOULD BE SUPPORTED BY A TEACHER MANUAL THAT IS WRITTEN IN PLAIN ENGLISH. IT SHOULD INCLUDE:
  - SAMPLE RUNS
  - COMMENTARY
  - STUDENT RECORD SHEETS
- FLEXIBLE FOR USE IN A VARIETY OF SETTINGS/LEVELS.
- FEATURE GRAPHICS AND SOUND TO MOTIVATE BUT NOT DOMINATE.
- FEATURE A COMBINATION OF SKILL BUILDING, CONCEPT DEVELOPMENT AND NUTURING OF LOGICAL THINKING OR PROBLEM SOLVING STRATEGY.
- "DEVELOPED" BY CLASSROOM TEACHERS.
- COST EFFECTIVE.

The most important characteristic(s) of good "games" cited by our teachers is that the activity should make a unique contribution by enhancing some skill, concept, or encouraging the development of a strategy. Games that lend themselves to use by individuals or teams are also rated quite high.

Some Suggestions for Game Authors

Because of the extensive time commitment needed to produce quality educational software, it may be useful and prudent to use a checklist suggested by a commercial software developer.

SOME SUGGESTIONS FOR GAME AUTHORS

GENERAL

- DO YOU HAVE IDEAS WHERE THE COMPUTER PLAYS A UNIQUE ROLE?
- ARE YOU WILLING TO GIVE THE TIME AND $ IT TAKES?

SPECIFIC

- STUDY GAME ROOMS FOR TIPS. (ALLADIN'S CASTLE)
- GET HELP BUT DON'T ABDICATE.
- MAKE YOUR GAMES EDUCATIONAL AND GENERIC. (ALLOW TEACHER FLEXIBILITY)
- GO BEYOND (SKILL DEVELOPMENT)
- CONSIDER COMPUTERIZING AN OLD FAVORITE.
- TRY YOUR NEW GAME(S) USING PAPER AND PENCIL FIRST.
- FIELD TEST YOUR IDEAS WITH VARIOUS AGE GROUPS.
- MAKE SURE YOUR GAME IS "GOOF PROOF".

A Few of Our Favorite Games and How We Use Them

There are many ways to use microcomputer games. One way is to assign individuals or small groups to work with particular programs for a short interval of time. Another possibility is to play one or more games with an entire class as in introductory, review, or maintenance exercise by using a large video monitor for classroom display. Some programs that are used may provide a change of pace from typical practice sessions. Others may be entertaining and motivational. Still others may nurture strategy formation and other problem solving skills. A few programs may provide all these desirable features. The programs listed below are some of the favorites of our teachers and students.

SOME SELECTED GAMES AND HOW WE USE THEM

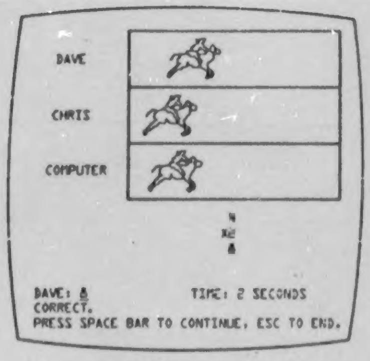
DISKETTE	GAME	OBJECTIVE	SETTING
MATH GAMES VOLUME 2	HORESERACE	BASIC FACTS	WHOLE CLASS (2 TEAMS - BOYS VS. GIRLS)
MATH GAMES VOLUME 2	MATCH UP TIC-TAC-MATH	MULTIPLE OPERATIONS COMPUTATION (W.N., D., I) NURTHRING STRATEGY FORMATION	SMALL GROUP (2 OR 3 PLAYERS)
MATH GAMES VOLUME 1	APPLE SQUARE-OFF	PLOTTING ORDERED PAIRS NURTURE STRATEGY FORMATION	WHOLE CLASS
MATH GAMES VOLUME 4	MATH DARTS WOODCHUCK	TIMED COMPUTATION (#s 0 → 100) COMPUTATION (#S 0 → 255)	SMALL GROUP OR WHOLE CLASS
MATH GAMES VOLUME 6	ASTEROID	PRACTICE USING >, < AND = (W. N., F., D., I)	LIBRARY/FUN STATION
MATH ACTIVITIES VOLUME 2	MATH GALLERY	TIMED MATH PROBLEMS (+, -, x, /)	
MATH ACTIVITIES VOLUME 3	TREASURE HUNT	TIMED MATH PROBLEMS (+, -, x, /)	
MATH ACTIVITIES VOLUME 4	PINBALL MATH	TIMED MATH PROBLEMS (+, -, x, /)	
MATH ACTIVITIES VOLUME 4	TOP OF THE MT.	WORDS TO NUMERALS	
MATH ACTIVITIES VOLUME 3	UNLOCK IT	PLACE VALUE AND LOGIC	WHOLE CLASS
MATH ACTIVITIES VOLUME 5	MIX UP	VOCABULARY PRACTICE	WHOLE CLASS

Horse Race

Objective: To provide timed, computation practice of basic facts for 1 or 2 players/teams.

Description: Graphics and competing against the computer are used to stimulate interest in this program. The participants may choose addition, subtraction, multiplication, division or a random mixture of problems. Users may set the ranges of the numbers. Figure 1 shows the beginning of a race.

Special features of the program include complete error proofing and optional music. If a student responds incorrectly the correct answer is given but his/her horse does not move.



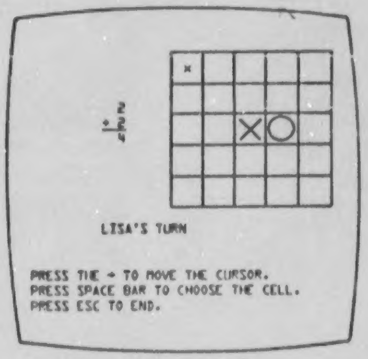
```
DAVE: 8                    TIME: 2 SECONDS
CORRECT.
PRESS SPACE BAR TO CONTINUE, ESC TO END.
```

Figure 1

Tic-Tac-Math

Objective: To provide practice adding, subtracting, multiplying or dividing using whole numbers, decimals or integers and nurture strategy formation for 2 players/teams.

Description: The object of the game is similar to ordinary tic-tac-toe. Players take turns solving problems. If the answer is correct the computer will allow the player to place his/her mark in one of the cells on the board. Special features of the program include a wide range of problem choices, to provide variety and adjust to various needs and grade levels. Another key feature of this program is that users may select the size of the board and the number of marks in a line required to win. (Figure 2)



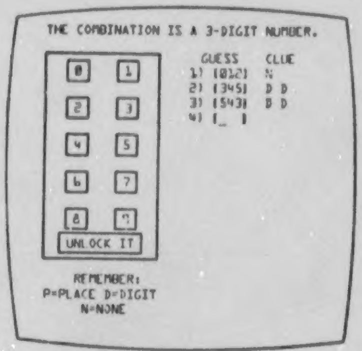
```
LISA'S TURN

PRESS THE → TO MOVE THE CURSOR.
PRESS SPACE BAR TO CHOOSE THE CELL.
PRESS ESC TO END.
```

Figure 2

Unlock It

Objective: To provide practice in developing place value understanding and logic skills using two-, three-, and four digit numbers.

Description: Figure 3 shows an activity as it could appear on a computer screen. The student/class must give the combination to an electronic lock. After each guess, the computer gives clues about the digits in the guess. Special features of the program include allowing users to select the number of digits, whether to enter their own codes or have the computer make up a code and whether to play for prizes.



```
THE COMBINATION IS A 3-DIGIT NUMBER.

      GUESS    CLUE
   1) [012]    N
   2) [345]    D D
   3) [543]    D D
   4) [_  ]

   REMEMBER:
P=PLACE D=DIGIT
   N=NONE
```

Figure 3

Conclusion

There is a growing body of action research evidence that indicates that adding computer games featuring student teams and competitive computer interaction to our repertoire of teaching techniques may be one of the most effective approaches to learning and problem solving yet devised. This seems to be particularly true when the computer game or activity features the use of fantasy and allows or encourages the student to develop a systematic strategy or plan. Try using educational computer games in your classroom(s). We think you will like the results.

References

Lund, C., Computer Math Games, Vol. 1-7, Addison Wesley, 1982
Lund, C., Computer Math Activities, Vol. 1-5, Addison Wesley, 1983

TECHNIQUES FOR INTRODUCING
BASIC IN GRADES 4 AND UP

Anthony Maffei
Pee Dee Education Center
Post Office Box 829
Florence, South Carolina 29503

Perhaps the appropriate time to introduce BASIC programming in the schools is around grade four. Programs have been devised that are of interest to this age level (and beyond) and are not too easy nor too difficult. These programs should be personally appealing, involve graphics and movement, and include a variety of commands. They are taught using a model as a guide in a problem solving setting. These programs are unique to those with TRS-80s and Apples.

Beginning programming experiences should be enjoyable as well as instructive. This can be done by introducing simple model programs which each demonstrate the variety of ways one particular BASIC command can be used. In addition to their conciseness and clarity, these model programs are personal, fun, stimulating, and graphic. Students then create their own programs using the model as a guide. This method can also be used in introducing teachers to programming.

The following show some common BASIC commands with model programs. Students first enter each of the model programs for a particular command. They are then requested to write their own program based upon the model program. Problem solving strategies evolve when students provide their own plans to solve these problems. Common underlying steps should be experienced by each student as they progress from command to command. As a result, their ability to solve other new problems is increased.

Some Examples

THE PRINT STATEMENT

Purpose: Will output on screen (CRT) any number or anything inside quotes.

A.

```
05  PRINT 1
15  PRINT 2
20  PRINT 3
25  PRINT "LET'S GO!"
```

NOTE: Pressing RETURN after each line you type will store that line into the computer's memory.

RUN   Type this word to execute your program.

B. NEW   Type in (erases previous program from the computer's memory).

```
10  PRINT "IT";
20  PRINT "LOVES",
30  PRINT "ME",
40  PRINT "NOT"
    RUN
```

NOTE: Semicolons after a quote put output together on the same line while the commas separate the output.

You Do These

Use PRINT statements in a program to create these types of output:

1.
```
      TRY                    TRY
      TO                     TO
   REMEMBER                FORGET
```

<u>HINT</u>: Try three PRINT statements and commas!

2. Print a rocket ship like this one!

```
        X
     X     X
     X     X
     X     X
       XXX
       XXX
```

<u>HINT</u>: You're on your own!

Just For Fun

TYPE and RUN these two different programs. Type in your name between the quotes.

A.
```
10  PRINT "         ";
20  GOTO 10
```

<u>NOTE</u>: Hold the CTRL key down and press C key to stop program.

B.
```
NEW
10  PRINT "         ",
20  GOTO 10
```

<u>NOTE</u>: See how the "," and ";" affect your output?

We will examine the GOTO shortly!

THE <u>GOTO</u> STATEMENT

<u>Purpose</u>: Will switch control of program to a certain line number designated in program.

A.
```
01  CLS
05  PRINT "I LIKE ME"
08  PRINT
10  PRINT "I AM GREAT BUT SLIGHTLY"
12  PRINT                   REPETITIOUS"
14  GOTO 12
```

(What is happening in Line 14?

B.
```
05  CLS
10  PRINT "HI MY NAME IS CLYDE COM-
          PUTER"
20  PRINT
30  PRINT "I HAVE A PROBLEM"
40  PRINT
50  PRINT "CAN YOU GUESS IT?"
60  GOTO 60
```

(Press BREAK key to end the run)

<u>NOTE</u>: We can "freeze" the run of a program with a GOTO to its own line (see line 60).

C.
```
10  CLS
20  PRINT @ 404, "A TWINKLING STAR"
25  PRINT @ 474, "*"
27  CLS
29  GOTO 20
```

<u>NOTE</u>: Why is the star blinking? What line(s) are causing it to blink?

You Do These

1. Get the computer to write your name repeatedly. (See Program A for an idea.)
2. Have the computer print your favorite TV program repeatedly somewhere down the middle of the screen. (<u>HINT</u>: Try commas for getting your output near the middle.)
3. Have computer blink near the center of the screen the name of the person you like the most. (See Program C for help!)

USING MACHINE READABLE SURVEYS: A PRACTITIONER'S PERSPECTIVE

JON C. MARSHALL
University of Missouri-St. Louis
School of Education
AFSE Department
8444 Natural Bridge Road
St. Louis, Missouri 63121

ABSTRACT

The session provides an analysis of the utilization of machine-readable surveys in education. The analysis includes looks at design time, formatting, administration, data handling, reporting information obtained, response rates, and costs.

INTRODUCTION

Discussed in this paper are a variety of issues that confront the users of surveys to collect educational information. These issues concern the proper design of surveys, development procedures, administration, handling of the data once it is gathered, reporting, and costs. A major focus in this paper is the interaction between each of these factors with the utilization of machine-readable surveys.

The discussion of these issues would be of interest to those individuals involved in obtaining information through survey techniques. Typical types of information obtained through this type procedure would be student follow-up, teacher concerns, parent feedback, and staff needs.

Much of the literature on survey development and administration is buried in the measurement and research methodology literature, with such issues as item writing and sampling. The common mythology is that a survey has to be under 10 items in length and one page in length. All items are to be of a multiple-choice type. And, so on.

Quite often, it seems, there is little concern for the specific contents of the questions, the response format, the data handling problems, or the decision-need for the data. In educational surveying the emphasis is often on the collection of data rather than on the analysis and utilization of quality data.

There is a need to determine a few guides for the development and utilization of surveys in education. This need is amplified in relation to the developing technology. Added to the other concerns is the learning how to utilize the technological developments to our advantage.

Some examples can be used to illustrate the concern.

EXAMPLE 1. A large suburban school district developed a student follow-up questionnaire as part of the school guidance program. The form was the normal one page survey with the appropriate 10 questions. It was mailed, with a self-addressed stamped envelope, to about 2,000 high school graduates. The questions were typical of this type survey.

- I WAS SATISFIED WITH THE JOB THE COUNSELORS WERE DOING.
- I ENJOYED MY YEARS IN - - SCHOOL DISTRICT
- I GOT A GOOD EDUCATION

While these questions are easy to respond to and the survey had to take only a few minutes to answer, the response rate was only six percent. Few students -- only about 120 out of 2,000 bothered to fill out the survey and return it. Moreover, the reponses provided nothing more than a general happiness index for the district.

****

NOTE: This example may seem extreme. However, not six weeks ago I read an article in the local newspaper that reported the results of a student follow-up study conducted by an area university. The questions quoted in the article were exactly of the type illustrated and the responding sample was only about 10 percent.

****

EXAMPLE 2. A second large suburban school district developed a community survey as part of an overall district evaluation and planning effort. This survey was designed to fit on a one-page multi-panel form. It was neatly conposed, using boxes for individuals to check their responses. It was folded in a z-form so that it fit into a regular #10 envelope that was self-addressed and stamped. The survey contained only six questions. However, most of the questions had multiple responses, for a total of 75 possible answers that could be marked by each adult in a household. Close to 20,000 surveys were mailed out. A nearly 30 percent return was obtained for a total of over 5,000 individual surveys. Now, the problem occured. How do you analyze the data? To prepare a timely report for the school board, the responsible administrator had her secretary hand tally the results of the first 400 surveys to be returned. From this tally, a couple of frequency tables were developed and a short report written. The school board rejected the report requesting a more complete analysis of the data. At this point, three secretaries were commendered for a week of keypunching. Spending about two-thirds of their time at this task, they made their way through about 1000 surveys. These were statistically analyzed using a mainframe. The data were broken down into several relevant categories for analysis, the report written and presented to the school board. Even though it represented only about one-fifth of the available data, it was accepted by the board. This example illustrates the problems that occur because of a lack of planning for data handling. All the efforts in survey development were wasted because of an oversight, a common oversight.

****

NOTE: Just yesterday I filled out and returned a survey from the school district within whose boundaries I live. According to the letter, the results were suppose to help with accreditation review. It was a commercial survey; not a homemade one. It ask important, relevant questions like:

- STUDENTS ARE RESPECTFUL TO THEIR TEACHERS (rating)
- TEACHERS SET GOOD EXAMPLES (rating)

At best these questions would be general happiness scores. The results would have little value in actual developmental efforts. The validity of the responses would be questionable since many of the items called for conclusions that could only be the result of considerable obsevation in the schools. Moreover, the survey consisted of a multi-page booklet that would require hand processing for analysis.

AN ALTERNATIVE

One method for developing and administering surveys that can alleviate many of the problems cited in these two examples is optical scanning. In making this suggestion, it is realized by the author that many individuals view optical scanning with great reservation, particularly with survey research. The coldness of requiring people to fill in little round bubbles with number 2 pencils on an obviously preplanned, somewhat sinister survey seems to strike many individuals as grossly insensitive.

For those persons feeling that machine processed surveys are cold and impersonal, two things can be stated. First, surveys in general tend to be a cold, impersonal method of collecting data, except, of course, when the information can be obtained through person-to-person interviews. Second, by necessity, machine processed surveys must be well planned. Not to do so will almost always end in complete disaster. It had been my experience that a much better return is obtained when using a well thought through survey that when using one that has been hastlly put together.

It should not be misconstrued that the development and utilization of a machine processable survey will solve all the survey development and utilization problems, or that it is necessarily the best method to obtain and handle data. Prior to deciding on a data collection method, the survey researcher should examine his/her data needs and options carefully. Several specific questions should be addressed when developing surveys.

1. *Purpose*. Is the survey to provide general perceptual information or detailed factual or evaluative information?

2. *Administration*. Is the survey to be mail administered, administered in a "captured" setting or through personal interviews?

3. *Design Time*. Do the surveys have to be ready in a few days or weeks or are there several months to develop the form?

4. *Formatting*. What type of information is needed? Are open-ended questions necessary or can selection questions be used? Is there sufficient time to format the questions in a tight space? Or, is time so short that it would be better to type into a booklet.

5. *Quantity of Information*. Are only four or five pieces of information needed or tens or hundreds of pieces of information needed?

6. *Number of Respondents*. Are there going to be 10, 50 or 100 repondents or are there going to be 500, 100 or more respondents?

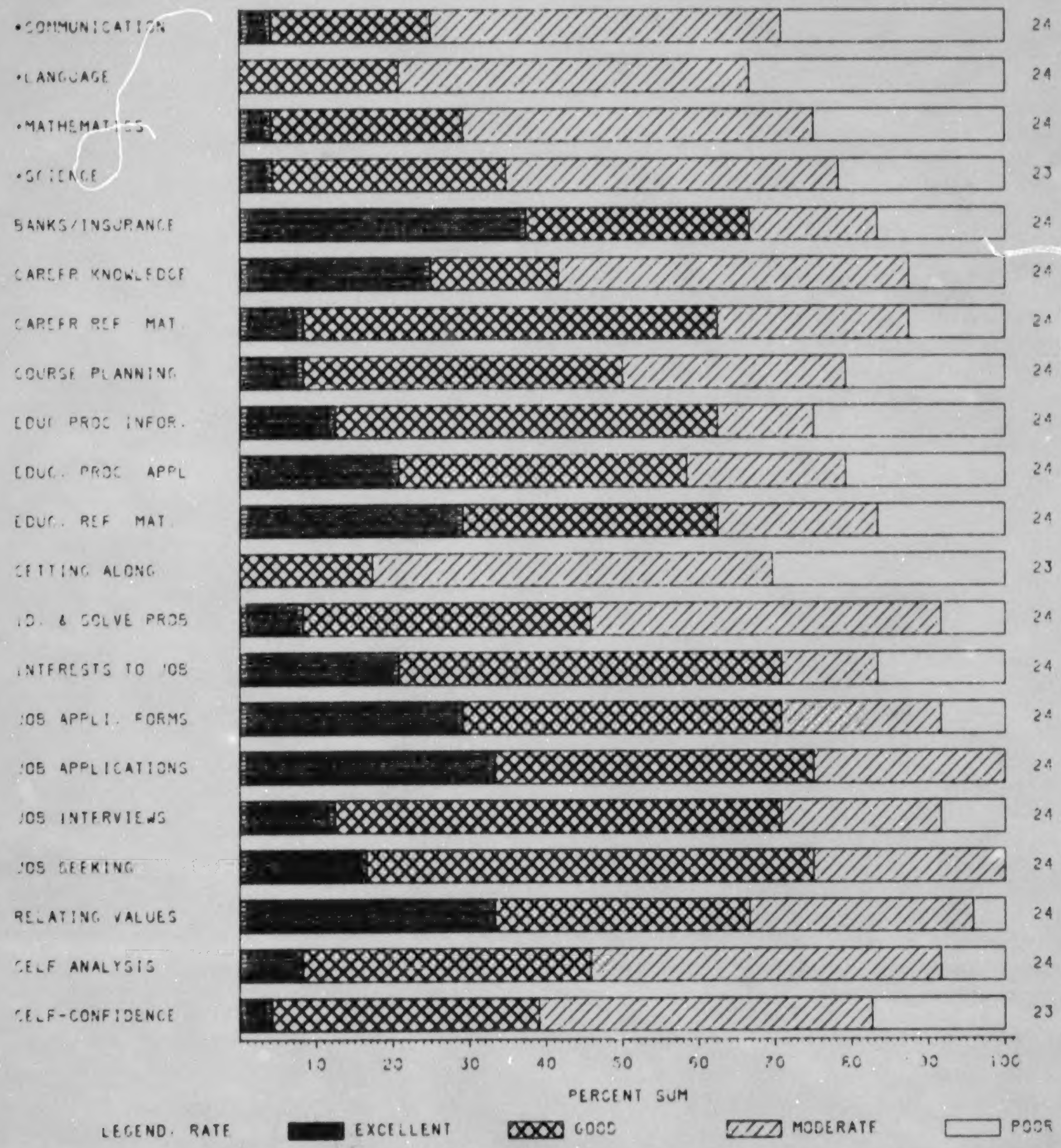
7. *Number of Administrations*. Is this to be a one time only collection of data or is it to be repeated on some continuing basis?

8. Data Capturing. Are the data to be hand coded for analysis or are they to be put on a computer for analysis and storage?

9. Analysis. Are simple frequencies of a few items to be reported or more sophisticated analyses to be performed? Are graphics to be utilized? (Note: See an example of computer produced charts from a student followup survey using SAS in the illustration below.)

10. Reporting. Does the audience want a short written report only or a more detailed statistical analysis?

11. Costs. How much money can and should be put up front in the development of the survey? What will be the maintenance costs if the survey is to be used more than once? What are the analysis costs? What are the cost trade-offs? If more initial money is put into the survey will total costs be less?



COMPUTERS IN EDUCATION - SOFTWARE CATALOGUING PROJECT

Gordon G. McKye
Etobicoke Board of Education
1 Civic Centre Court
Etobicoke, Ontario, M9C 2B3

ABSTRACT

The purpose of this paper is to describe a unique co-operative project by 28 different Boards of Education in Ontario that has resulted in a catalogue of 900 public domain PET programs. The programs themselves are not new; rather they are drawn from the body of public domain programs for the PET that teachers all over North America have been using for the last two years. What is unique, however, is that every program has been technically improved to a uniform standard so that it runs and incorporates at least some elements of a technically sound program. Furthermore, the programs and the catalogue of those programs are available to anyone who wants them.

A word of explanation concerning the Ontario Software Cataloguing Committee

The Computers in Education Software Cataloguing Committee resulted from an informal meeting in October 1981 of a number of teachers and consultants who had been involved during the summer of 1981 with a variety of student projects funded under various government projects. The majority of the summer projects had involved the improving and cataloguing of public domain software for educational use on microcomputers. There was a strong feeling at that time that a co-ordinated cataloguing effort should be undertaken immediately in order to provide some structure for further summer projects and to co-ordinate the efforts of various boards. Since no group or organization seemed to be involved at that time, the group of people present decided to continue to meet under the name "Computers in Education Software Cataloguing Committee". Gordon McKye, Co-ordinator for Computers in Education for the Etobicoke Board of Education, agreed to chair the committee.

Since no jurisdiction had sufficient time or resources to proceed with an Educational Software Catalogue, it was decided to ask the Ministry of Education for support to fund an initial cataloguing project. With the help of the Ministry, a teacher was hired to sort and catalogue existing public domain software for the PET microcomputer. The result of that project was that, by June 1982, a catalogue of 1200 unique computer programs was compiled with programs sorted on disks according to subject and grade level.

Standardized Cataloguing System

In preperation for the cataloguing part of the project, other members of the committee devised a standardized cataloguing system which would be used for cataloguing computer based resources. Agreement was reached on the information to include and on the way the information would be coded. Each catalogue entry would include the following data:

1. Program title
2. Subject headings - science, math, language, etc
3. Second subject cross reference
4. Disk id - the programs are compiled alphabetically within each of the 16 subject area and the disks are assigned a source id.
5. Category - drill, game, simulation, tutorial, utility, other
6. Grade level - early child, primary, junior, intermediate, senior, college, trainable mentally retarded
7. PST vector analysis - 3 digits representing the presentation, structure and tracking characteristics of the program
   First digit - presentation
   - 0=page turning
   - 2=input of single alpha or numeric
   - 4=alpha/numeric manipulation
   - 6=non interactive graphics
   - 9=interactive text/animated graphics

   Second digit - structure
   - 0=non progressive
   - 2=problem level progression
   - 4=branching/remedial
   - 6=variations under teacher control
   - 9=variation control by students interaction

   Third digit - tracking
   - 0=non tracking
   - 2=marking for program only
   - 4=cumulative marking
   - 6=marks compared to class
   - 9=marks compared to external standards
8. Status - public domain, copyright, licenced
9. Computer - PET/CBM, APPLE, TRS, ATARI
10. Memory size - minimum memory size of computer needed to run program
11. Author
12. Source
13. Reviser
14. Special requirements
15. Annotation or comments - limited to 108 characters

Standardized Program Improvement

Again, coincident with the cataloguing aspect of the project, a subcommittee devised a series of useful programming subroutines. The purpose of these subroutines was to provide a basis for a uniform provincial project to improve the software that was being catalogued. A manual was prepared to accompany the subroutines to show student programmers how to incorporate the subroutines into an existing program. Once incorporated, these subroutines provided a menu based selection routine, a selective input routine, disable of the run/stop key, uniform page design, a unique help function with an automatic return to the appropriate place in the program, plus several other routines. It must be understood that these subroutines are not perfect and they do not please everyone. Over the course of the project, we have found many improvements. Nevertheless, in June of 1982, these subroutines were agreed upon as forming the basis of the uniform improvement that would be imposed on all programs.

The subroutine package with documentation was provided to 28 different student projects (representing 28 different Boards of Education) each under the direction of a teacher or consultant. Each project was given some group of programs from the original 1200 catalogued programs. Each group was asked to screen the programs to make sure that programs had educational value and then to perform the improvement routine which included making sure the program ran properly, had adequate instructions and incorporated the subroutine package. It must be stressed that evaluation and pedagogical improvement was not the purpose of the initial project. The programs were not evaluated by an educator. Rather the initial project was to take a large body of public domain software, select programs of general educational value, make them technically sound and make them available to boards and teachers as they find useful applications in the classroom.

Distribution of Program Disks

The combined result of the efforts of the 28 groups is a body of 900 programs that are available for use by any interested person. The programs are stored on 79 different program disks. The programs may be copied and shared in any way desired, since, to the best of our knowledge, they are in the public domain. The 900 programs were made available (at cost for disks) to the 28 boards that participated in the project.

Since it is a very time consuming task to make copies of so many disks, it was felt inappropriate to ask the participating boards to make copies of the programs for other interested boards or individuals. At the same time it was recognized that the disks will be of value to many teachers and should be made generally available. Two small groups have been given the task of copying all programs (by whole disk only) and making them available at minimum cost. If you are interested, please write to one of the following:

Aurora Software,
P.O. Box 1394,
Haileybury, Ont, P0J 1K0
or
Computertown Canada,
c/o Windsor RCSS Board,
1485 Janette Avenue,
Windsor, Ontario, N8X 1Z2

Ask for the "Ontario Software Cataloguing Project - Catalogue of PET Software". You will receive by return mail a list of available disks and can then order what you wish.

Commodore Business Machines has taken the 900 programs on this project and has converted 600 of them to run on the new Commodore 64 microcomputer. These are available from Commodore dealers in Canada and the USA.

A new 1983 Ontario Software Catalogue of Public Domain PET Programs

The committee which started the original project met on December 6, 1982 to consider the usefulness of the contribution made to date and to consider what additional potential exists for co-operative action. It is clear that the catalogue has made a significant and useful contribution across the province and beyond. The most pressing criticism is that there is no evaluation of the usefulness or accuracy of the programs and there is need for considerable pedagogical improvements on most programs. Another need recognized by the committee concerns the updating of the present cataloguing effort. Many additional programs were not included in the original project but are now available. Often these programs are written by educators and are better than many programs already in the catalogue. A third need expressed by teachers in Ontario is for a similar collecting and cataloguing of programs for computers other than the PET.

The committee is satisfied that there is room for continued co-operative action and has decided to pursue the above needs in the following ways. First, a project is underway to have all of the programs in the present catalogue evaluated by at least three teachers. The catalogue information will be updated by the teachers and a written critique of each program obtained, specifying the changes that are necessary in the program for its acceptance as an educationally useful resource. The committee has prepared an evaluation form for the purpose of collecting this information. The task will be divided by assigning different groups of programs to each site willing to undertake evaluation. The requirement is that at least three qualified teachers will view each program. The catalogue listing in the printed Ontario Software Catalogue must be verified as correct or changes specified. Lastly, a detailed description of any changes required in the actual program must be provided. These changes must be specified in sufficient detail and clarity so that a capable student working in the summer can implement the changes.

A number of groups across the province will be using summer student assistance under various government grants to take part in this project. Each group will receive a number of programs and will be responsible to implement the necessary program changes as specified by the reviewing teacher. Following implementation, all programs will be returned to a central depot for organization into a form that can be distributed back to participating boards. It is anticipated that the 1983 edition of the Ontario Software Catalogue of Public Domain PET Programs will be somewhat smaller than the 1982 edition but of greater educational value.

Catalogue of TRS-80 and APPLE software

The committee has not neglected the other two challenges. New programs that are currently not included in the catalogue will be collected, catalogued, and evaluated. It is hoped that these programs can be evaluated and also be included in the improvement process planned for this summer.

Another project concerns software for other micros. A group of teachers has indicated an interest in starting a project to collect, catalogue and improve public domain software for the TRS-80 microcomputer. The committee will support this effort. If there are teachers interested in a similar project for the APPLE software, a similar project could be started.

GETTING STARTED WITH LOGO: A DIRECT INSTRUCTION APPROACH

Samuel K. Miller

ABSTRACT:

Logo is an immensely popular programming language that is being touted as the most significant educational software of the decade. Its simple commands allow both children and adults to create complex graphics and manipulate lists of numbers, words and symbols. Logo is regarded not only as a programming language, but as a philosophy of learning sometimes referred to as a "Logo environment." No formal program for creating a Logo environment has existed until just recently. This paper is intended to present the rationale for the development behind Getting Started with Logo, an instructional program for Logo-using teachers and students.

A logical place to begin an examination of the program is with the Logo philosophy itself. The person most responsible for the development of Logo and its philosophy is Dr. Seymour Papert. Papert, a mathematician by training, studied with the famous Swiss psychologist Jean Piaget. The Logo philosophy is based upon Papert's interpretation of Piaget's theories of how children learn, particularly the idea that children are the builders of their own intellectual structures. Diciples of Papert regard the ideal environment for teaching Logo as the development of both a physical environment and a set of ideas which, when used properly, will allow the creation of truly intellectually independent students. The most common portrayal of a teacher in a Logo classroom is that of a facilitator for discovery learning experiences. No specific objectives are formally measured for the purpose of grading students. Students in a Logo environment are usually not required to complete specific projects. A Logo environment has been conceptualized as a place where process is emphasized instead of product. In short, the teacher in a Logo environment does a very limited amount of direct instruction.

The idea for creating an instructional program for introducing Logo to teachers and students occurred to Ron Thorkildsen and me while we were both doctorial students at the University of Oregon. Both of us had a strong interest in educational computing and experience in developing curriculum materials. Ron was familiar with the Logo language because it was an integral part of the "Computers in Special Education" course that he taught at the University. My own experience with Logo included teaching inservice courses and instructing gifted elementary students how to program in Logo.

Ron and I would frequently discuss our observations about teaching Logo to children and adults. We came to some common conclusions that eventually formed the basis for the development of the program Getting Started with Logo. The program, published by the Developmental Learning Materials Corporation, consists of a teacher's guide, activity sheets, and instructional posters for Apple Logo. In summary, the rationale for the program is predicated on the following ideas:

1. Directly teaching students important information about the Logo programming

language and reviewing that information does not compromise the philosophical integrity of a Logo classroom.

2. Students can benefit from structured activities that present them with models from which they can extend in a myriad of ways.

3. Teachers frequently express the need for printed support materials that will provide them with a sense of where to begin instruction.

The following sections briefly describe some of the teaching strategies and instructional design considerations may be adapted to Logo classes that are not currently using the program.

TEACHING THE LOGO LANGUAGE

The commands and syntax rules of the Logo language do not constitute the essence of the Logo philosophy. The Logo language is simply a "key" that will allow the user to "open" the computer for use as a thinking tool. Fortunately, the Logo language is friendly enough to establish a good first impression with students; however, this does not preclude the necessity for introducing the language in a systematic manner. Ideally, a Logo classroom would include a number of computers and a group of students capable of being self-directed. In reality, most schools have a limited number of computers and students display a wide range of abilities. One objective of Getting Started with Logo is to meet the needs of teachers with little or no background in teaching the Logo language, especially under circumstances that are less than "ideal". The basic format for introducing the Logo vocabulary in the program is as follows:

1. The teacher directly presents new commands and concepts that progress from simple to complex. For example, it is more efficient to introduce students to variable procedures with one input, followed by variable procedures with two inputs. The first concept can be used to help explain the second, more complex idea.

2. Whenever a new command is introduced it is modeled by the teacher in the context of a wide range of examples. If the REPEAT command is being introduced the teacher would demonstrate a variety of ways that REPEAT could be used. This approach is necessary for teaching young children and handicapped learners, because these learners frequently fixate on one specific use of a command. It is not uncommon to see the naive Logo using youngster "superstitiously" using commands in certain sequences or patterns because they have internalized some implied misrule about how they think a command must be used.

3. At some point in Logo students' experience, it will be helpful if the students can recall commands and syntex rules from memory. However, until students become facile with the use of the Logo language, they should be provided with charts and other reference materials.

4. Teachers must be cognizant that when they introduce information about Logo they must be careful not to present students with too much information in any one lesson. Teacher talk should be limited and student participation must be maximized. For example, students can be provided with simple fill-in sentence sheets that can be utilized as notes while information is being presented. This is a simple way to actively involve all students during a lesson.

5. Frequent oral questions during a presentation are an excellent way for the teacher to verify whether or not students understand information that is being presented. For example, the teacher might ask the class, "Who can help me define a procedure for drawing a tiny box using the REPEAT command?"

Written and oral exercises that review information about Logo are useful insofar as they promote the idea that information that has been presented is worth remembering (otherwise it shouldn't be taught).

6. Error messages from the computer may sometimes shake the confidence of a Logo-using student. Teachers must demystify error messages in advance. An important theme in a Logo environment is that errors are an acceptable and natural part of problem-solving with computers. It is extremely beneficial for teachers to demonstrate and explain what will occur when a command is not presented in the correct syntax to the computer. These examples can be used to generate short, high utility rules to present to the students. For example, "To make the turtle move, type a command followed by a number." The teacher could then demonstrate how the Logo language responds when someone types the command FORWARD with no input number. When Logo responds, "NOT ENOUGH INPUTS TO FORWARD," the teacher can then explain that inputs means "put in a number with the command."

USING MODELS TO TEACH LOGO CONCEPTS

An important objective in a Logo environment is for students to manipulate that Logo language to explore and discover what Seymour Papert refers to as "powerful ideas." An example of a powerful idea is that of combining separate procedures to create a new procedure. Teachers cannot assume that "modular" thinking is an automatic part of every student's problem-solving repertoire. Some important Logo concepts can be presented via models that the student can use to generalize information for solving similar problems in the future. For example:

1. Whenever a new Logo command is introduced, the sutdents should have an opportunity to practice using the command. A teacher prepared worksheet containing sample procedures and drawings is a useful way to demonstrate various applications of a new command. It is imperative that students be taught that the examples do not represent the only way to solve a particular problem. Students must be provided with examples that demonstrate that there are usually a variety of ways to solve a problem.

2. Complex ideas such a combining procedures, how coordinate commands function, and recursion must be prefaced by a carefully worded teacher presentation. This presentation should be followed by student activities prepared by the teacher. For example, a presentation about recursion might be followed by student activities such as: (1) examining and copying models of recursive procedures; (2) transforming non-recursive procedures into recursive procedures; and finally, (3) "suggestions" for independent projects that require the use of recursion.

PROVIDING SUPPORT MATERIALS FOR LOGO-USING TEACHERS

Some Logo experts would say that by its nature, Logo is better implemented free from the constraints of a suggested sequence, activity sheets, and recommended teacher activities. Such assertions inevitably require individuals to affiliate themself with a particular educational "tribe". Educational tribes include Skinnerians, Piagetians, and the "non-aligned." Each tribe espouses a theoretical position that usually succeeds in polarizing groups of people.

The instructional "philosophy" put forth in this paper can be classified as the Direct Instruction approach. This does not mean that the authors of Getting Started with Logo believe that the teacher should direct students through every activity. It simply means that teachers and students can benefit from the presentation of information and materials that will enable them to rapidly learn important preskills that are necessary for solving problems with the Logo language.

Many of the important considerations in the development of the written materials for Getting Started with Logo were derived from information provided by teachers involved in field testing the program. Some of these considerations are listed below:

1. To provide teachers and students with explanatory materials about the Logo language that are: (1) free from technical jargon; (2) carefully sequenced; (3) rich in graphic examples that illustrate various concepts; and (4) written to provide a presentation format for teachers who were teaching Logo for the first time even if only one computer were available.

2. To provide teachers with activity sheets that could be utilized both on-computer to: (1) model and reinforce information about Logo commands and concepts; and (2) provide ideas for projects that would require students to devise original solutions.

3. To provide teachers with a compendeum of resources for teaching Logo that includes: (1) strategies for teaching students how to devise a plan, record it and debug it; (3) off-computer activities that reinforce geometric concepts, practicing Logo commands, and learning how to correctly orient the turtle on the screen; and (4) information about publications and organizations dedicated to exploring new ways to use Logo.

Based upon our experience in training teachers and students, these were the components of a Logo support program for Logo-using educators, at this point in the history of educational computing, to build a synthesis of educational philosophies that succeeds in selecting the strengths of each view of education to capitalize upon the strengths of that marvelous mind-tool--LOGO.

COMPUTER BASED LEARNING
IMPORTANT PROBLEMS, CREATIVE PEOPLE, AND POWERFUL AND AFFORDABLE TOOLS

Kenneth L. Modesitt
Texas Instruments Incorporated
P.O. Box 10508, M/S 5890
Lubbock, Texas 79408
(806) 796-3463

ABSTRACT

Computer-based learning (CBL) is compared and contrasted to classical learning systems. Before creative people can work on important problems, they must be provided affordable tools. CBL is one such powerful tool. It is not, however, a cure-all. Therefore, we must be aware of needs assessment, human competence, learning, and finally, computer-based learning. Some advantages of CBL will be presented, followed by a discussion of the use of CBL at Texas Instruments, both as a third-wave tool for society at large and as one for internal productivity.

COMPUTER-BASED LEARNING

One model for learning is given below, with both classical (circa 1000 A.D.) and electronic candidates.

Needs	Production System	Information	Network	Input/ Output Device	Peripherals
			CLASSICAL LEARNING SYSTEM		
school curriculum	professor at desk with paper and pencil	lecture notes gradebook	walk to classroom	professor	blackboard chalk
			LEARNING SYSTEM USING TECHNOLOGY		
needs assesment	hardware software authoring	courseware manageware	telecommunications	terminal	VTR
			POTENTIAL PRODUCTS		
organizational elements model	CDC Cyber PCD2 TI authoring aids	PLATO courses maintenance	Bell system DBS cable	TI home computer CDC 110	video disk video tape lab eqpmt

FIGURE 1: Model for Learning System

The use of high technology in computer based learning (CBL) from computers to communications is very much evident. The use of the dotted line (----) in figure 2 denotes infrequent network access, just on a demand basis, whereas the solid line denotes fulltime hookups.

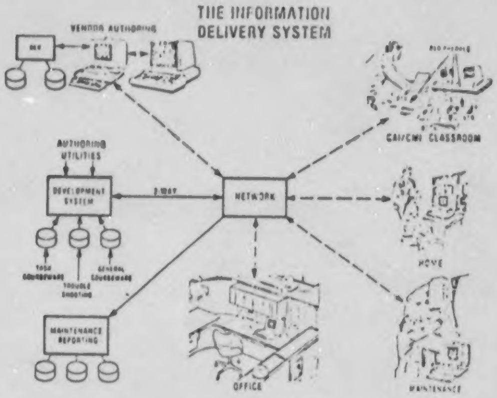


Figure 2. Technology-Based Learning with Network

However, both of these concentrate on the hardware related aspects of a CBL system, and do not specifically address the content nor the style of the material. The content is so broad that not much meaningful can be said. "Name three areas which could not use CBL...." On the other hand, the style is of considerable interest to both the author and learner. So some definitions may come in handy.

Tutor: the student is tutored by the computer executing a program constructed by a person (or team consisting of subject-matter expert, instructional designer, graphics artist, and programmer --the best of all possible worlds in the view of some). This is the domain of "computer-assisted instruction" or "CAI": drill and practice, dialog, simulation, inquiry, tutorial.

Tool: the computer has some useful capability programmed into it such as word processing, calculator, statistical analysis, graphics

Tutee: the student tutors the computer by communicating procedures and/or problems to it in a language it understands.

Computer-managed instruction: diagnosis, prescription, remediation, testing, tracking, manage other learning resources

Authoring system: all the facilities of a production system required to generate learning materials, from concept to delivery

Authoring language: a programming language (from very primitive to very high level) which permits an author to construct a lesson.

Courseware library: existing curricula, courses, modules

Network: electronic mail, note files, lesson comments, on-line consultant

One of the favorite pastimes of "experts" in CBL is to argue over the relative merits of large versus small computers for learning systems. A recent article gave several salient features and then compared them for both types of systems.

Features	Standalone (micro/mini) PASS, PILOT, SIMPLER BASIC, LOGO, SMALLTALK	Timeshared (maxi) PLATO, IIS, AIS PHOENIX
Portability	+	-
Response time	+	+-
Hardware installation	+	-
Small training operations	+	-
Impact of one system failure	+	-
Collect response data	-	+
Progress reports	-	+
Vendor support	-	+
Control sequencing	-	+
Costs		
purchase/unit	+	-
development	-	+
operations	+	-
maintenance	-	+
Author base	+	+
Courseware base	-	+
Authoring system	-	+

(after Fauley and Dennis, 1980)

FIGURE 3: Comparison of Existing (partial) CBL Systems

It should be clear that a hybrid system is the most desirable, wherein a "hybrid" column would be all "+"s. This would combine the advantages by using standalone operation primarily, but with easy access to large networks for lesson delivery and on-line conversations and help. The large network would be used for author resources, management and people interfaces. The standalone mode would be used for quality and inexpensive delivery. Everyone does not require a printing press in their home, yet they are strongly desirous of the results thereof.

CBL AT TEXAS INSTRUMENTS

Texas Instruments Incorporated has always had a well-deserved reputation as an innovator for society, especially in the use of electronic tools. So the work we are doing in various divisions involving computer-based learning will be very much available to society.

One of the most exciting ventures currently underway involves the hybrid CBL system mentioned earlier. This involves the coupling of the TI 99/4A as a delivery device for the PLATO computer-based learning system of Control Data Corporation. The TI computer provides:

0 low cost
0 quality
0 light-weight and portable
0 color graphics and animation
0 multi-sensory (speech, sound)
0 expandable
0 video interfaces
0 wide-spread consumer distribution

This delivery device requires an interface for generating quality interactive courseware cost-effectively. The recently signed contract between TI and CDC provides just such a proven tool. The PLATO system is known worldwide for:

- largest single investment in CBL over last twenty years
- current user base exceeds 150 corporations and 80 universities
- 15,000 hours of CBL materials, with an annual growth over 2000
- breadth of learning materials for K-12, university, government
- comprehensive managing of individual learning with multi-media

Details on the mechanics of the technical challenges will appear in a later paper. It should be clear that this is precisely the "hybrid" system discussed earlier. PLATO courseware on the TI home computer will have virtually all the advantages of both the large computer resources for authors as well as inexpensive quality delivery to millions of people world-wide in their homes, schools, and businesses.

There will be difficult times in bringing all these efforts to fruition, as the problems of mass merchandising such CBL tools is a new venture in some respects. Selling millions of such devices poses many intriguing challenges. Nonetheless, we are committed to products which are:

- easy to use
- affordable
- of high utility in important markets
- marketed via normal distribution channels
- compatible

SUMMARY

Learning is a very potent process in our world, as we struggle to become more self-sufficient, interdependent, and contributing members of our societies. It is not a cure-all, as there are often many alternatives of reducing the gaps between our "what is" and the desired "what should be". When chosen wisely, learning can open broad new vistas. Technology, when chosen wisely, can provide effective and remarkably efficient tools for the learning process. This application, under the name of computer-based learning, is proving to be one of most exciting, challenging, and comprehensive ventures taken on by our institutions, including universities, governments, industries, military, K-12 schools, and now the home. Creative people can indeed be found everywhere across the world. Provide them with easy-to-use, powerful, and affordable learning tools and "stand back" as we hop aboard the "third wave".

REFERENCES

1. F. Fauley and V. Dennis, Pushbutton Training, Training and Development Journal, 34, pp.110-114, 1980.

2. T. F. Gilbert, Human Competence: Engineering Worthy Performance, McGraw-Hill, New York, 1978.

3. T. F. Gilbert, Training: the $100 Billion Opportunity, Training and Development Journal, 11, pp. 3-8, 1976.

4. G. Kearsley, Authoring Systems in Computer Based Education, Communications of the ACM, 25, pp. 429-437, 1982.

5. R. Kaufman, Identifying and Solving Problems: A System Approach, Third Edition, University Associates, San Diego, 1982.

6. W. Sick, Meeting America's Technical Skills Shortage: Electronics Technology in the Classroom, Address to Chief State School Officers, Columbus, Ohio, 1982. Reprints available from Texas Instruments.

7. R. P. Taylor (editor), The Computer in the School: Tutor, Tool, Tutee, Teachers College Press, Columbia University, New York, 1980.

8. "Technology", Gould Incorporated, 10 Gould Center, Rolling Meadows, Illinois 60008.

9. A. Toffler, The Third Wave, William Morrow and Company, New York, 1980.

PROJECT DISCO

Susanne Murphy

Eastern Junior High School
Hendrie Ave.
Riverside, Conn. 06878

ABSTRACT

This paper describes a pilot project whose purpose is to demonstrate the use of the laser videodisc in interactive courseware. Superpilot is the authoring language used, with Apple micro-computers, a Disco-Vision Model 3 laser player and an Allen Communications VMI interface board.

A brief background discussion of laserdisc technology, its potentials and drawbacks is also included.

INTRODUCTION

Computer technology has influenced education in a variety of ways; although technologists are inclined to contend that the process has been glacial and reluctant. Educators, on the other hand, are quick to point out that the early efforts of programmers in curriculum design more than justified the teachers' initial distrust of technological panaceas. Clearly, there is right on both sides: education does change slowly, and programmers are not likely to be both subject matter experts and learning process specialists as well as facile coders.

Into this mix of mistrust and ignorance, technology inserts a further complication. Development in both hard and software makes possible techniques which both corporate and school house educators could use if they were aware of them. How does high tech get out of the lab and into the mainstream where it can be used, evaluated and changed? This paper describes a project which attempts to do just that.

BACKGROUND

The laser videodisc is a medium for recording, storing and playing both linear TV-like segments and still-frame pictures. There are two different formats of laser disc and several kinds of player available today. As yet, laser discs are not erasable and re-recordable, although indications are that they soon may be. Superficially, the comparison between laser disc and videotape would seem to favor tape. Tape is less expensive to produce, erasable, and its attendant machinery is much more widely available and cheaper. Laser discs, on the other hand, are few, exotic, and expensive. Laser players are not widely available in classrooms or industrial training sites. Two significant differences which tip the balance in favor of disc, however, are interactivity and random access.

The laser videodisc player at the high end of the market is an intelligent device with a microprocessor built in. The user is able to interact with the player to control speed, order and format of the presentation. More importantly, control programs can be stored directly on the disc, loaded into the player's memory, and used to individualize the presentation for a variety of viewers. Depending upon user input, the same disc can give many different presentations to different users. Ford Motor Company has developed a training disc which responds to user input by branching to different portions of the disc. The University of Nebraska Tacoma Narrows disc, now marketed through John Wiley and Sons, New York, allows a user to experiment with physics and engineering principles on a variety of learning levels. These high end, "intelligent" players are also fast enough in their random access so that the annoying delays of videotape systems are virtually eliminated.

Very recently a further enhancement to interactivity has become available. Interfaces now exist which marry the conventional micro-computer and the videodisc player. Our project makes use of the VMI Board produced by Allen Communications. Now much more elaborate control of the videodisc is possible because of the larger memory available in the micro and because of the ease of user input associated with the standard computer keyboard rather than the remote control button devices of the laser player. The very popular authoring language, PILOT, has also been expanded into SUPERPILOT which includes commands for laser player control.

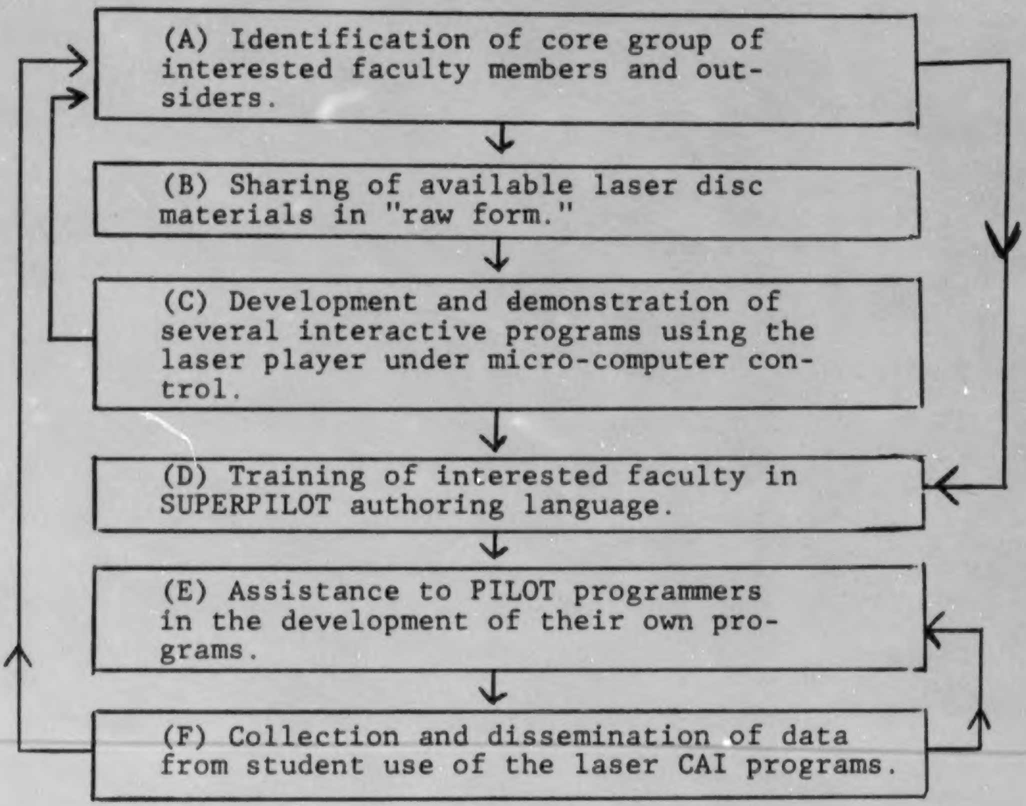
What we have, therefore, is a technological tool of great power available to educators, corporate trainers, video producers, archivists/librarians and data base planners alike. These are the capabilities and potentials; to a great extent they exist very much still in the laboratory.

PROJECT DESCRIPTION

One needs, initially to justify the use of the technology itself. The battles over computer-assisted education and "computer literacy" have by no means yet been won. It is possible that the laser disc and authoring languages will help greatly to strengthen the case for computers in the home, workplace and classroom. On the other hand, one needs to be very clear about which skills computers can and cannot teach effectively. We have overall far too little research data on which learners are most appropriate for CAI and which least.

What we have chosen to do is to work with those already interested in, and, (at least in some sense), commmitted to, computer-assisted instruction. We have a fairly extensive computer center for a junior high school (10 Apples) a progressive, competitive faculty, a sophistocated population base, reasonable (but it's never enough), financial backing, and a middle-of-the-road approach to educational innovation. The plan for the laser videodisc project is diagrammed in Figure 1.

Figure 1


(A) Identification of core group of interested faculty members and outsiders.
(B) Sharing of available laser disc materials in "raw form."
(C) Development and demonstration of several interactive programs using the laser player under micro-computer control.
(D) Training of interested faculty in SUPERPILOT authoring language.
(E) Assistance to PILOT programmers in the development of their own programs.
(F) Collection and dissemination of data from student use of the laser CAI programs.

Of the core group of interested persons, many are using the laser discs in "raw form." That is, they are showing the linear TV shows as they would an educational videotape. The music department has used the "Music Is" series, math teachers, "Infinity Factory," and social studies and science people, the "NEA School Disc" and "Tacoma Narrows" respectively. Foreign language and English-as-a-Second-Language staff have made greatest use of the interactive and authoring aspects. The "Villa Alegre" series is particularly exciting to this group. Special education and remedial classes, with their more individualized curricula, make more use of all technological aids including the videodisc.

We are now at stage 'D' on figure 1.

CONCLUSIONS AND FUTURE PLANS

1. Everything takes approximately twice as long as anticipated.

2. Laser disc technology will come into its own as an educational tool only when interactive programs are available commercially. "Villa Alegre" which is complete with laser and computer discs, is much more popular than the "do-it-yourself" discs.

3. Most teachers have very limited amounts of time available for interactive program development. What is needed is a supply of "ready-to-go" materials with well-written instruction manuals. It would be unrealistic to suppose that classroom teachers will learn languages (either programming or authoring) in very great numbers or very soon.

4. On the other hand, school systems (and publishers!) could profitably devote time and money to the production of interactive curriculum materials. Teachers are eager to make use of the technology;and children love it.

Based upon our experiences with this project, Greenwich will probably move in these directions:

1. Laser players with micro-computer interfaces available in each school.

2. Authoring languages available in media centers.

3. Staff development programs will include courses in the authoring language(s) and interactive curriculum design along with computer literacy and BASIC.

4. Media aides and other supportive staff will be trained to assist teachers in the development and/or selection of the appropriate laser disc teaching/learning aids.

5. The central media services center will begin to acquire a circulating collection of laser disc materials along with its 16mm. films, videotapes, computer programs, etc.

GEOPROCESSING AS APPLIED TO AN INTEGRATED DATA BASE

Ian M. Neelan
Portland Public Schools
501 N. Dixon
Portland, Oregon 97227

ABSTRACT

This paper relates the experience of Portland Public Schools in merging a Geoprocessing system with an integrated data base. The emerging applications for educational institutions from this merge are described.

The paper begins with a description of Geoprocessing and Integrated Data Base technology. The relationships between bus routes, students and geographic base data are shown using PPS's data base diagram. Experiences with transportation routing and boundary change applications are related and other applications are mentioned. The paper ends with a discussion of future applications such as interactive graphics.

INTRODUCTION

Portland Public Schools is a relatively large district with a current enrollment of 51,000. Our schools are organized into basically 3 different grade types, primary schools K-5, middle schools 6-8 and high schools 9-12. There are 7 Early Childhood Education Centers that offer pre-K classes. These centers are open to city-wide enrollment. There are also magnet programs at several of the high schools which are also open to city-wide enrollment. We have a total of 95 regular schools and 23 special schools covering an area of 152 square miles. There are approximately 14,000 students being transported to and from school at the present time including 1400 special ed students and 4000 administrative transfer students many of which are participating in our voluntary desegregation program. We use 250 buses for this purpose 75 of which are district owned and 175 contracted.

Inflation and declining enrollment created a need to close schools despite community opposition, provide more mandated services to increasing numbers of minority students and reduce costs. With the increasing transportation complexities and the need to reduce budgets the Data Processing Department began looking for a computer assisted transportation system. The results were surprising in that the software package found dealt not just with transportation, but with any application that could be associated with geoprocessing.

GEOPROCESSING

Geoprocessing may be defined as the coding, storage and retrieval of geographically identified data. This data must be identified by means of points such as coordinates or geographic areas, such as census tracts, blocks or land parcels. If coordinates are used, a file of street segments can be created by using these coordinates as the endpoints. This is the basic concept for the census DIME (Dual Independent Map Encoding) file. Each record on this file represents a street segment whose endpoints are identified by state plane or world spherical coordinates. Other pertinent data such as census tract, census block, zip code, fire and police information is also present for each side of the street.

Portland Public Schools has acquired a geoprocessing system from Ketron, Inc. that is based on the DIME file. The main applications of this system are bus transportation and student reassignment. The system's concept is to match each student's address to a variant of the DIME file so that each student can be identified by a set of coordinates affixed to their home. These students can then be grouped into any kind of geographic area, such as school boundaries. Student geographic distributions can be displayed based on any kind of data that is retained on the student master file. The transportation system takes these geographically identified students along with a set of stops, also identified by geographic coordinates, and creates a set of bus routes. These routes after manual review are then "tiered" or assigned to the least number of buses possible. This allows fewer buses to serve the same number of routes. For this reason, Portland Public Schools is saving over 1 million dollars per year in transportation costs.

This system consists of approximately 110 COBOL programs generally in the range of 35k 36-bit words. Groups of these programs can be run interdependently or modified to provide different types of information.

All data files must be processed by some kind of text editor. However, with an Integrated Data Base and a screen generator, data updates can be quite easy and user oriented.

INTEGRATED DATA BASE

An Integrated Data Base can be described as a data file where each record is organized into logical set relationships, so that data can be accessed by various pathways.

Portland Public Schools uses a Honeywell DPS 8 mainframe with an IDS (Integrated Data Store) II Data Base. This data base makes use of set relationships in which there is an owner and a member. Fig. 1 illustrates these relationships. Each owner can have many members and each member can have one owner within a set. Pointers are attached to each record to maintain this relationship. Owner records have first and last member pointers. Member records have pointers that relate to the next and prior members of the set as well as the owner of the set. This allows processing of the set in either direction starting at any record of a set. An important advantage of this design is that one record can provide access to all related records within the data base. There are three different methods of entering this data base. The first uses what is called a calc key where the key is a data field specified by the user. A calc key is indicated by an arrow pointing into a record at the top left corner. The second method involves just finding the first specified record in the data base. The third is to search through all the occurences of a specified record finding a particular data field within these records that corresponds to a given piece of data.

ROUTE DATA BASE

I will first describe the application of bus routes to an Integrated Data Base. The organization of bus routes has been determined by the Transportation Department of the Portland Public Schools. The data base can be made to mimic this organization. First, a line is a set of routes that are run by one bus. Each line therefore can be said to own a set of routes or each route is a member of a line. The data base representation of this is shown in Fig. 2. Line records can contain data relating to the actual bus such as capacity and maintenance data. Also, each route is a member of a school or a school owns a set of routes. This along with the previous representation is pictured in Fig. 3. As shown the route records have two owners, but the set of routes for each owner can be different. The route record tells whether the route is an arrival or departure sequence and what days the route is run. A bus route contains a sequence of stops. Therefore, a route owns this stop sequence and a stop sequence is a member of a route. This can then be added to the previous part of the data base, pictured in Fig. 4. The stop sequence record details the time of the stop and the distance to or from the school. There are a set of stops within the district determined by the Transportation Department that can be used many times by different routes. Therefore, a stop contains a set of stop sequences and a stop sequence is a member of a stop. This is shown in Fig. 5. A stop record contains the address and coordinates of the stop. Since stop to stop directions pertain only to stops, these records will be attached to the stop records as shown in Fig. 6. As noted, the stop-to-stop record has two set pointers coming from the stop record. This is known as a network. In this case each stop owns a set of stop-to-stop records and each stop-to-stop record is a member of two stops. Below the stop-to-stop record is the stop-to-stop detail records, which are the actual set of directions for each stop-to-stop. Under each stop sequence record there is also another record type which represents the set of comments that can be attached to each stop sequence. The full route data base, therefore, is shown in Fig. 7.

A route list can be provided by walking through each set within this data base. Also other reports concerning routes can be created on request. For example, if a person needed to know all the routes that arrive and/or depart from a particular school, that person could find a school record directly through a calc key and find all the routes it owns. The same can be done for particular stops as well.

GEOGRAPHIC DATA BASE

The Geographic Data Base can be organized similar to the organization of street segments on the DIME file. This data base is shown pictorially in Fig. 8. The central record of this data base is the address range record, which defines one side of a street segment. This record is owned by a street record and a block record which in turn is owned by a census tract record. The address range record also owns three separate records dealing with stops, schools and node points or coordinates. The use of type records as shown is necessary since there can be more than one stop, school or node attached to the street segment.

With this data base we can find all the blocks in a tract, all the street segments within a tract-block, all the street segments in a street or all the street segments for a particular school, stop or node. If applied to a graphics package this data base can give a large range of pictorial mapping data. This could include school boundary and stop boundary maps.

COMPREHENSIVE DATA BASE

This data base is a joining of the two data bases mentioned above. In addition we can now add student data. This new comprehensive data base can be seen in Fig. 9. Students records can be attached to the Route Data Base through the stop sequence records by way of a transportation record that details the students reason for transportation as well as dates of entering and leaving the service. In this way reports can be created to show students with their line, route and stop numbers. Bus passenger lists are easily created by using this new connection. Students can also be attached to the Geographic Data Base through a date record, for history purposes, to the actual address range record in which the student lives. This allows us to create lists of students within school boundaries or any other boundary that can be defined. Also each student can be shown with the distance from his/her nearest stop or school. Determination of eligibility for transportation purposes can be accomplished more easily with this distance calculation ability.

FUTURE APPLICATIONS

For the future, update and edit functions on this comprehensive data base can be done using a set of screen and on-line transaction processing. Data brought together in the data base can then be shown on a screen and updated or edited immediately.

Graphically, a whole new world of pictorial data is now available to us. Using a plotter, student geographic distributions can be made based on grade, ethnic code or any other piece of data associated with students.

With a CRT, digitizer and a good graphics package, interactive graphics is a real possibility. One of the most useful applications of interactive graphics for Portland Public Schools would be to show school boundaries on a CRT and change those boundaries with a digitizer or light pen noting the student populations within the new boundaries. Updating and editing of the DIME file also could be done with this method. Another application would be to show the sequences of stops in a route on a street map.

Probably the least expensive method is to use a microcomputer for the interactive graphics. With a good communcations package, geographic data from the data base on the mainframe can be downloaded to the micro in segments. With a micro graphics tablet and micro graphics software an expensive CRT and digitizer could be avoided. A good plotter, however, is necessary for good hardcopy graphics when used for reports to management or school boards.

CONCLUSION

Information generated by a geoprocessing system can be enhanced by application to an integrated data base as well as providing on-line capability. Changes and additions to the data base can be made immediately. Managers can see the effects of hypothetical changes in school boundaries on a timely basis. Portland Public Schools has benefited greatly from this system in reduced transportation costs and in the availability of crucial information for the Board of Education, the Superintendent and his staff.

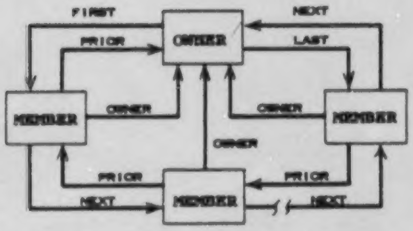


FIG. 1

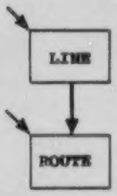


FIG. 2

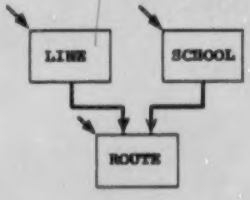


FIG. 3

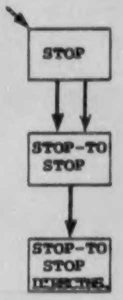


FIG. 6

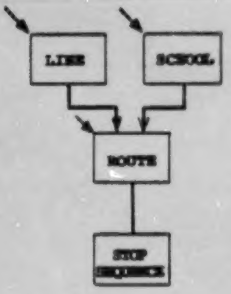


FIG. 4

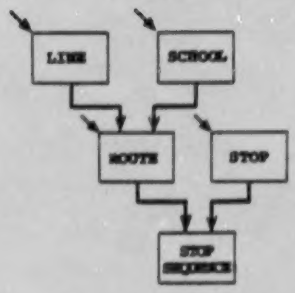


FIG. 5

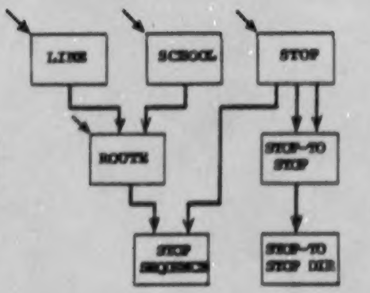


FIG. 7 ROUTE DATA BASE

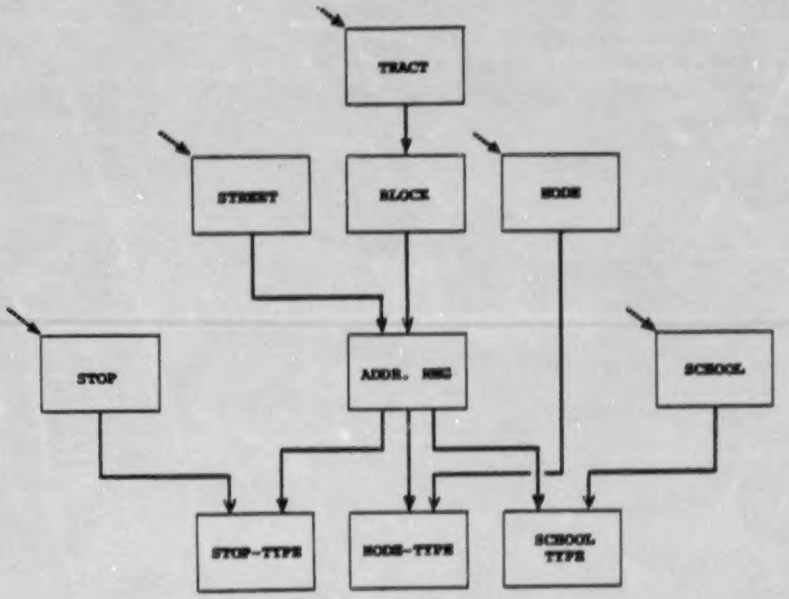


FIG. 8 GEOGRAPHIC DATA BASE

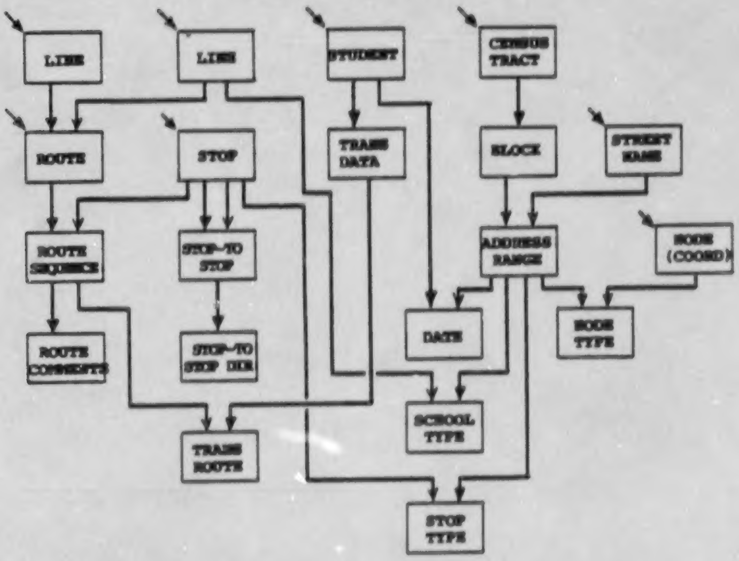


FIG. 9 COMPREHENSIVE DATA BASE

ECONOMIC ISSUES IN THE MANAGEMENT OF INSTRUCTIONAL COMPUTING RESOURCES

Norman Nicolson
Cerritos College
11110 Alondra Blvd.
Norwalk, Ca 90650

Schools and colleges are experiencing an escalation in demand for computing resources for instructional purposes at the same time that education is suffering from severe financial constraints. This paper will identify and discuss key economic issues confronted by those who are attempting to reconcile supply and demand at the organizational level. Questions to be considered include the "appropriate" level of instructional computing for a school or college, alternative approaches for effectively allocating computing resources among users, and issues of centralization vs. decentralization in the support of instructional computing.

INTRODUCTION

Many of us involved in instructional computing in schools and colleges are finding out how Oliver Twist must have felt when his humble and respectful request for "more" precipitated the Beadle's apoplectic fury that he wasn't satisfied with what he had already been given. School Boards, Superintendents, and College Presidents are being confronted with a dilemma of considerable magnitude, and of vital importance to the future of their institutions as the organizations entrusted by our society to transmit knowledge from generation to generation. On the one hand it is clear that we are now entering what many call "The Information Age", and for schools to provide relevant education will demand extensive utilization of appropriate computer resources: on the other hand it is painfully obvious to those with fiscal responsibilities that public education is facing a financial environment that calls for retrenchment. Some suggest that since our survival as a competitive industrial state depends on adequate investment in educational computing, the issue is one that calls for national commitment. However, the probability of significant federal action at this time seems slight, and responsible decision-makers in school districts and colleges have to make resource allocation decisions on an ongoing basis. In a budget-development era characterized by layoffs, salary reductions, and the repeated deferment of plant maintenance, just how much of an educational institution's budget should be spent on instructional computing? "More", some of us say, "much more". "We can't afford it" say the Business Managers and Boards. "We can't afford not to", we insist. Later in this paper we will attempt to develop some approaches to help decision makers to establish what is a reasonable or appropriate instructional computing investment for their institutions, but first let us look at some of the characteristics of computing in an educational environment.

THE INSTRUCTIONAL COMPUTING ENVIRONMENT

Most instructional computing activities carried on today are in the interactive, as contrasted to the batch, mode, and we can expect that this will be the case for virtually all instructional computing in the near future. Interactive computing may take place either with a terminal connected to a mainframe or a minicomputer in a timesharing environment, or with a standalone microcomputer. We propose that in discussion concerning instructional computing that the concept of an individual workstation should replace any debate over whether terminals or microcomputers are more appropriate for student use. The interactive workstation will serve as both; it will have a capability for extensive local storage and processing and will also be capable of high speed communication with mainframes and minicomputers, external sources of information and specialized computing resources, and other workstations. Indeed, this distributed information processing network can be expected to become a major learning resource with profound effects on the schools and colleges of the future. Instructional computing may be usefully categorized into those activities involved in learning about computers (primarily in computer literacy, programming, and computer science courses), the use of the computer as a medium of instruction in a multitude of curricular areas (e.g. computer assisted instruction in music or geography), and the use of the computer in ancillary activities independent of the particular subject matter involved. This latter category could include, for example, text processing, computer-based testing, or the use of computer mail communications among students and teachers. Clearly, a very wide range of instructional computing activities are currently going on and an even greater variety seems likely in the future. Before attempting to develop estimates for institutional computing demand, it is appropriate to consider some questions on the nature of equipment, and access to computing resources.

GENERAL-PURPOSE OR SPECIAL-PURPOSE COMPUTING?

It is not unusual for a college or school district to inventory its computing equipment and be quite surprised both by the amount and variety of hardware that it has accumulated. Often pieces of equipment, particularly microcomputers, have been acquired by individual schools or departments to meet specific, single-purpose, instructional needs (e.g. for teaching word processing or for computer-assisted instruction in geography) with little or no consideration given to their possible use for other purposes. While there may be circumstances when such an acquisition is cost-effective for the institution as a whole, it is more likely that we have an example of sub-optimization, decision-making in the short-term interests of some part of the organization at the expense of the larger institution. Sub-optimization in instructional computing is often manifested by equipment under-utilization and by a failure to exploit economies of scale (in purchasing, maintenance, supplies, software support, training, etc.) that would be inherent in standardizing on a limited variety of equipment. To carry this concept further, while instructional computing may appear superficially to have little in common with administrative data processing, there may be opportunities for effecting short-term and long-run savings by integrating all decision-making about uses of information-processing technology within the organization. While much is being said these days about the desirability of selecting software to meet one's specific need, and then buying the hardware to run it on, we are suggesting here that almost the opposite is desirable if one wishes to minimize total organizational costs. Rather, buy the most general-purpose hardware possible, even if it is not quite as appropriate for a specific task as some other equipment: the computer should be purchased as a general-purpose information-processing tool, not, with few exceptions, as a special purpose tool.

ACCESS TO COMPUTING RESOURCES

It is desirable that we deploy our equipment and develop allocation procedures to ensure that we are utilizing our limited instructional computer resources to the maximum educational advantage. Idle equipment benefits no one; it is vital that we get the most use possible at all times of day and throughout the year. Generally this will occur in an educational environment if we bring as much equipment as possible together in an open-lab situation. A number of smaller labs may have some psychological advantages, and save students some steps, but the additional overhead in securing and supporting them, and particularly the higher probability of under-utilization in such environments (based on simple queueing theory), strongly support the argument for fewer, centralized, labs. The open-lab approach permits higher utilization than scheduled labs, where equipment sits idle while lecture activities take place and when faster students have completed their assignments. Another concern in optimizing student access to computing resources, is ensuring that each student gets a "fair share" of time. Experience over many years with timesharing environments in higher education has shown that a small number of students tend to use the lion's share of computer time, unless adequate controls are put in place. Where some kind of chargeback mechanism is not possible or appropriate (although we strongly recommend exploring this approach), then rationing of one kind or another must be implemented to prevent misuse of expensive resources.

ESTIMATING DEMAND FOR INSTRUCTIONAL COMPUTING

Clearly, the total demand for instructional computing at a school or college is a function of the many demands from individual students and instructors. The particular resources demanded will depend on the instructional activity taking place (e.g. some users will need extensive processing, others will require access to large files or databases, some may need special-purpose peripheral equipment) and the amount of computing needed will fluctuate with time of day and time of year. For purposes of this discussion, however, a useful metric for demand is simply the hours per week that a student interacts with a computer-based workstation, and as discussed above, this will in most cases be a standard workstation for the institution. Weekly student contact hours (WSCH) will vary considerably from course to course, with possibly less need for hands-on work for an introductory computer literacy class than for a course in advanced programming. It should be a responsibility of the instructional staff to define the appropriate WSCH for each current and planned course, recognizing that in some cases the figure may represent a mean WSCH per student, with actual usage for individual students ranging across a statistical distribution of some kind. The aggregation of these course demands, adjusted to reflect realistic scheduling and other problems, will provide a partial measure of overall institutional demand for instructional computing. To this, however, may or may not be added some additional demand factor, not related to specific courses, but reflecting school or college committment to providing general-purpose computing to students.

GENERAL-PURPOSE COMPUTER ACCESS

Reference was made earlier to student use of computing resources in activities that are not directly concerned with any particular course. How much computing should be made available to students for such general-purpose uses? This depends a great deal on the role that computing plays in education, as perceived by policy-makers for the school or college. Richard Van Horn of Carnegie-Mellon University has suggested that institutions may select strategies appropriate to their view of computing in the instruction process. The lowest level strategy might be to consider computing as an elective which some students may choose to study; the next level is a "literacy" strategy (every student should attain some level of computer literacy). Then comes a "skill" strategy, which views computing as a universal intellectual skill similar to natural languages or mathematics. We might think of the strategies as representing levels on a hierarchy concerning both the role of computing and some measure of computer utilization per student. The top of the hierarchy would be the adoption of a "foundational" strategy, which recognizes computing as the key technology for information processing and information processing as the fundamental process in learning. An environment operating under this strategy would imply students having daily involement with computing for up to several hours per day. We suggest that each school district or college invest the time necessary to establish explicity its philosophy on instructional computing, using Van Horn's, or some similar, hierarchy. Based on this, the institution can then develop plans for providing an appropriate level of computing resources, over and above that required to support specific courses.

SUMMARY

We have suggested that before expending additional budget on instructional computing each school district, college, or other educational institution examine carefully the utilization of its current resources. It has been claimed that, in general, most effective usage will occur in large clusters of workstations available to all students in an open lab situation, and that efforts should be made to standardize on hardware and software due to significant economies of scale. The fundamental unit of instructional computing demand is the weekly student contact hour, and by aggregating course-by-course WSCH figures we can estimate one component of total institutional demand. The other component depends on what the organization perceives as its appropriate instructional computing strategy, and it is suggested that developing this strategy is a vital activity for every school.

STAFF TRAINING FOR SUCCESSFUL CAI

Bruce Page
Systems Specialist
Salem Public Schools
P. O. Box 12024
Salem, Oregon 97309

Cherie Wallig
Library Media Coordinator
Salem Public Schools
P. O. Box 12024
Salem, Oregon 97309

ABSTRACT

To assure a successful implementation of Micro Computers for computer assisted instruction one must do many things, two of which are planning and training. Planning and training must be integrated to assure a person is trained for the next step of the implementation. We have identified four different groups: classroom teachers, school administrators, "building computer experts", and interested community groups. Each group has different training needs. This paper will detail the content of each training program and tell how it was implemented.

INTRODUCTION

In the spring of 1982, one elementary school and one middle school each received four Apple computers. This was the beginning of a pilot program in computer assisted instruction and Salem School District's entry into the world of micro-computers. It is now one year later. Sixteen of Salem's 37 elementary schools, 4 of 6 middle schools, and 2 of 5 high schools have Apple computers that are being used for CAI. In this short period of time it has been necessary to train teachers, direct implementation programs, and provide a support system for software use.

THREE-PART PROGRAM

First, an 8 hour inservice was developed. This was offered in the summer of 1982 and has been presented periodically through the school year, on a first-sign-up first-served basis. In addition, a special series of 8 hour inservices was held for the teachers from schools who purchased computers. Another type of presentation was prepared for parent groups who were interested in funding computers for their school and wanted to know what computers were all about and how they could help children learn. A third aspect of computer training began in January, 1983, in the form of a 30-hour class in BASIC programming.

The second area of district support was to provide direction for creating a computer program in the school. In order to avoid having the computer become a toy, discarded after the novelty wore off, an implementation plan was developed that would assure staff involvement, curriculum integration, and a feel for the commitment required to make the program successful.

Third, to provide schools with material to begin their CAI programs, the district purchased all of the MECC software and made it available for both preview and purchase by schools. A catalog of materials was printed and distributed. A previewing system was created using the current district AV materials circulation system. If schools find the material appropriate and choose to purchase it, copy of the disk and documentation is packaged, cataloged and processed, and sent to the school to become part of the media collection. Additionally, commercial software is brought into the district for three media exhibits during the year. Material is evaluated and lists of recommended software are distributed to the schools. This, coupled with professional reviews in journals, provides a basis for selective purchasing. The exhibit also provides teachers an opportunity to broaden their experience with software and to be aware of new software developments. Following is a brief description of the implementation plan and a detailed look at the inservice training program.

IMPLEMENTATION PLAN

Principals of schools purchasing computers received a printed packet of information at a two-hour inservice. This meeting also included the following presentation:

1) principals of the pilot schools reviewed what they had tried and learned so far;

2) the director of curriculum explained our focus on curriculum integration;

3) the computer program coordinators described the inservice program and software system and answered questions about the implementation program.

The packet included
- -an overview of the pilot programs, including 10 key elements identified as essential to program success;
- -a review of how the implementation plan should be used and a work plan format to follow;
- -a description of the computer inservice;
- -a summary of considerations for selecting software, plus directions for previewing/purchasing materials within the district;
- -a list of resources, including people, periodicals, and books;
- -material to be used when training students to use the computer, such as basic directions for running programs and caring for diskettes;
- -sample scheduling systems.

The projected result of using the implementation plan is that schools will have programs with clearly defined objectives and a systematic process for achieving them through the use of computers.

TRAINING PROGRAM

When we started evaluating our training needs we identified four groups: teachers, administrators, "building computer experts," and the interested public. We soon realized that the teacher and the administrator had the same needs. We combined these groups leaving us with three distinct needs.

Before designing the training courses, we set some basic ground rules:

1. All courses would be of a "hands on" nature.
2. We would do the training in one of two computer labs in the district.
3. We would use the CAI stations, which consist of an Apple II Plus, a disk drive and a color monitor.
4. We would concentrate on computer assisted instruction, and not yield to the pressure to teach about recordkeeping systems or student programming.
5. We would use software already available in the district.
6. The number of participants per computer would not exceed 3.

TEACHERS

The most pressing need was obviously to train our teachers and administrators. During the 1981-82 school year a few of our schools were able to purchase Apples. In those schools in which planning for implementation had been done and staff trained, computers were being integrated into the curriculum. The others seemed to be floundering.

Our training of teachers and administrators started during the summer of 1982. We brought into a lab setting 10 Apple II Plus computers. The computers were used from 3 -5 p.m. daily for instruction. The rest of the time they were available to staff to work on. Many days you could not get a computer to work on because the lab was full. Currently the training is held from 4-8 p.m. on weekdays.

The first session of the course is an introduction to the Apple. It consists of:

- -a very brief verbal introduction
- -15 minute slide/tape show produced by Salem schools introducing basic micro-computer concepts
- -discussion of some computer terms
- -a look inside the Apple pointing out such things as RAM, ROM, power supply
- - using Apple's keyboard demo
- -booting several disks

Each participant must go through all of the keyboard demo. Letting one person of a team dominate the keyboard at this point will lead to problems later.

The second session is on programming. The objective is to develop an appreciation for the logic and precision needed to write computer programs as well as understand and use the Apple system's commands.

The system's commands Run, Catalog, Home, List, Lock, Unlock, Load, New, and Save are discussed at the beginning of the session. Later in the session, they are used.

The programming part of this session is progressive. The instructions Home, Print, End, Input, Goto, If-then, and For-next are introduced one at a time. The participants key in a line of code, then run it. After the participants have seen the result of each line, they go to the next.

The concepts of string variable, numeric variable, and random number are also addressed in the program. The program upon completion will: title the screen, ask student's name, print student's name, ask number of problems to be attempted, ask greatest factors to be used, display 2 factors, ask for product, give positive response if correct, ask if operator would like to run program again. We are seldom able to complete the program in 2 hours. However, this is not a concern because the most important concepts are done first.

At various points along the line the participants are required to back up their programs. If they do not, as a result of an error, "wipe out" their program, the instructor will be sure it gets destroyed at some time during the session. The purpose of this is to demonstrate the need for backup of programs and information.

The third part of the computer inservice focuses on evaluating CAI materials. The three main types of programs - drill and practice, tutorials, and simulations - are explained. The developmental process of software production is discussed, leading to information on the MECC system.

The following factors are defined and class members use MECC materials to find good and bad examples of each:

- appropriateness for intended audience
- directions
- visuals/graphics
- ease of use
- motivation
- feedback
- accuracy
- adaptability/user control
- use of computer capabilities
- value/curriculum match
- documentation

Some examples of commercial software are then previewed using the district evaluation sheet that contains the above factors, plus specific questions on grade level, type of student, and type of learning.

The fourth part of the inservice deals with integrating CAI into the curriculum. The programs developed by the pilot schools are discussed with emphasis on identifying a target group, writing objectives, and scheduling. Research on the effectiveness of CAI is reviewed. Teachers are introduced to other uses of the computer in schools, such as Logo, literacy, programming, problem solving, word processing, and management systems; the rationale for these is compared to CAI. Teachers are encouraged to brainstorm about how CAI can work in their own class setting, using specific examples from MECC materials when possible. The expectations for teachers who complete this eight hour inservice are the following:

-they will feel comfortable using the computer and will be able to share their knowledge with students.

-they will have an awareness and appreciation of the structure of computer programs.

-they will be familiar with several MECC programs and understand how to integrate them into the curriculum.

-they will be aware of the factors that determine the quality of CAI materials and be able to use them when evaluating/selecting software.

-they will feel motivated and competent to participate in the development of a computer program in their schools.

INTERESTED PUBLIC

The training session designed for the interested public was set up for 2 hours. After a few minutes of discussion of micro-computers in general and Apple computer in particular, the participants are placed in front of a computer and told how to boot them. The Apples boot to the MECC demo disk and the instructor gives instruction 'on the run' between machines. Most of the people get through most of the MECC demo disk in about 45 minutes.

The MECC demo disk inevitably brings up some questions. These can usually be answered in about 15-20 minutes. Then it is back to the Apples to look at more MECC software which would have been chosen especially for the group based on their interest.

BUILDING EXPERT

The training for the "building expert" is a 30-hour programming class for which the participant can get 3 hours of credit. This class also uses Apple II computers and the language taught is Applesoft. The emphasis in the course is on developing CAI programs. A great deal of time is spent on graphics with emphasis on color and sound.

There are 3 students per computer in the programming class and each participant is expected to work on programming a few hours a week outside of class time. This has not caused any problems because one of the requirements to get into the class is access to an Apple in school or at home.

Our classes are being very well accepted by our teaching staff. As you might anticipate, our problem is not a shortage of teachers who will volunteer their time to take computer classes, but rather a shortage of computers and trained staff to teach the teachers.

CONCLUSION

The success of the CAI program depends on three main elements. First, the training sessions provide the participants with KASH: Knowledge, Attitudes, Skills, and Habits. The habits become established through the second element, the monitored implementation plan. The plan by itself will not create a program. It needs to be carried out. The district audits the implementation plan using the theory that you get what you inspect, not what you expect. Third, the provision of software and a delivery system offers the district some control over how the computers will be used. These three factors have allowed Salem to begin the integration of computerized learning into the existing curriculum.

IDENTIFYING HARDWARE
for
HIGH SCHOOL MICROCOMPUTER LABORATORIES

Charles C. Philipp
Montgomery County Public Schools
Rockville, Maryland 20850

The number of microcomputers used for instruction in high schools is increasing. Moreover, classroom applications are becoming diverse, ranging from the formal study of computer science to the use of pre-made software by business education students. One cost-effective strategy to provide hardware is to establish a multi-purpose computing laboratory.

The focus of this paper is a set of techniques that may be used to specify the characteristics of hardware required to support a variety of high school courses. Emphasis is placed on using and/or modifying the tools of structured analysis.

Use of microcomputers to support a variety of subjects in high schools is a new phenomenon. Until recently, most computing activity took place in mathematics and science classrooms. Equipment was limited to a few machines and the investment was modest. Developing interest in computer technology has, however, created a high level of demand for access to hardware. This paper is about a process that is currently being used to identify hardware for computer laboratories in Montgomery County, Maryland, Public Schools (MCPS).

At the present time, plans call for computer labs in eight high schools for use starting in the 1983-84 academic year. Each lab will be configured with 15 student work stations. After all 20 high schools have such labs, the number of machines per school will be increased to 30. The labs will support pilot offerings of three courses: Introduction to Computers, Programming for Problem Solving and Computer Applications (a course designed to teach students to use word processors, data bases, spread-sheets and similar software packages).

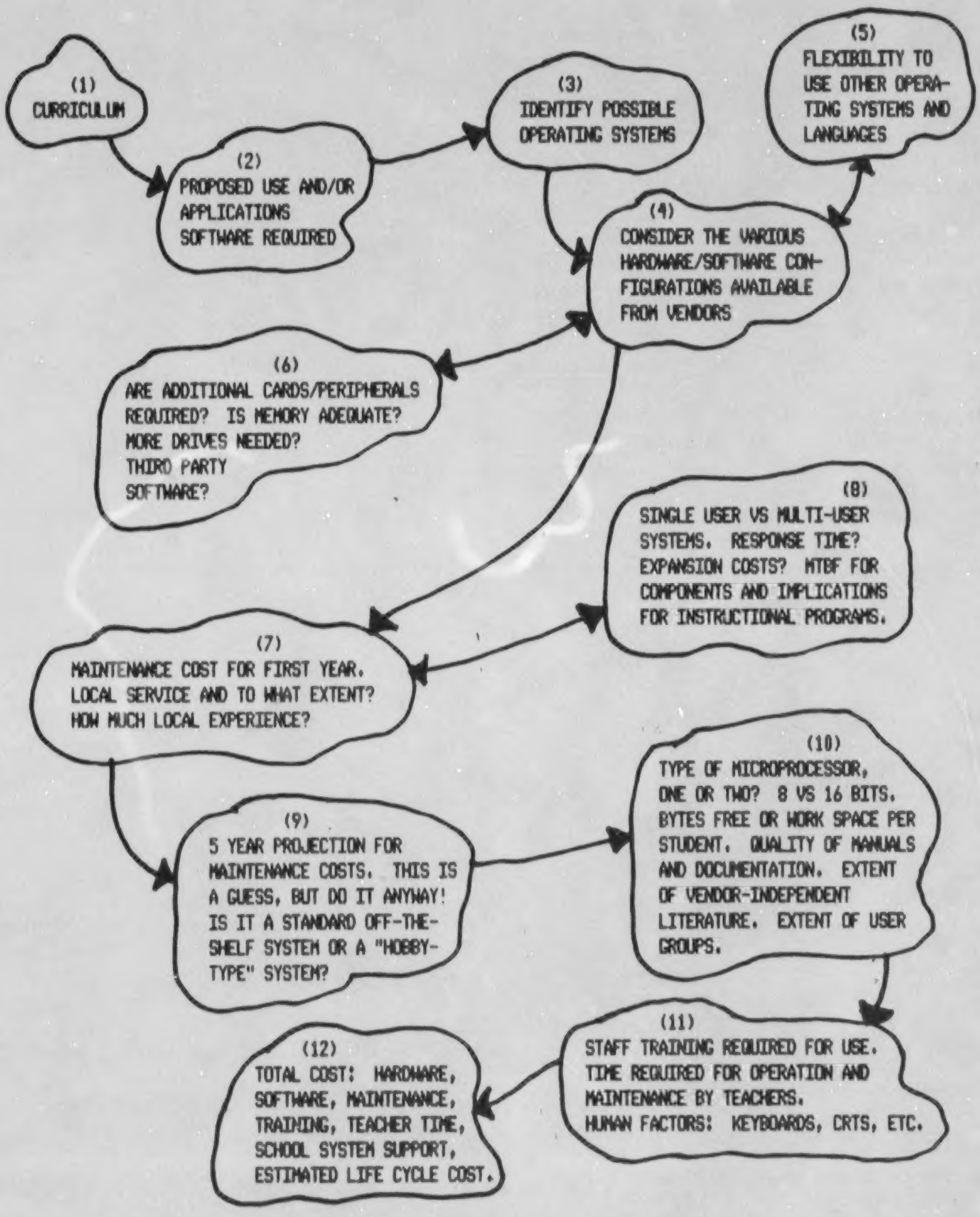
In addition to the above, three semesters of advanced course work involving Pascal are planned for the 1984-85 school year. Teachers of business education, science, mathematics, journalism, English, and social studies are also expressing interest in using computers for instruction. As curriculum is developed and teachers are trained, the labs will provide opportunities for computer use in a variety of subjects.

An effective framework for identifying hardware must be useful to those charged with this responsibility and make sense to administrators and teachers who are interested but do not know much about computers. A framework should also set technical standards and be connected clearly to the proposed instructional program for students. To the extent that it is possible, jargon should be avoided. There are, however, some technical concepts that must be included. Furthermore, the framework should be presented in the form of a diagram or picture that provides for linking the components of the process and allows individuals to see why it is sometimes necessary to consider a number of different parts simultaneously. Figure 1 is a diagram of the process now being used in MCPS.

STEP 1: CURRICULUM The starting point for the use of computers in classrooms must be curriculum. This ought to come before any discussion of hardware. A logical case must be built to show that computer technology will serve a legitimate instructional purpose. Interviews with potential users (teachers and supervisors)

A 12 STEP PROCESS TO IDENTIFY HARDWARE

FIGURE 1



(1)
CURRICULUM

(2)
PROPOSED USE AND/OR APPLICATIONS SOFTWARE REQUIRED

(3)
IDENTIFY POSSIBLE OPERATING SYSTEMS

(4)
CONSIDER THE VARIOUS HARDWARE/SOFTWARE CONFIGURATIONS AVAILABLE FROM VENDORS

(5)
FLEXIBILITY TO USE OTHER OPERATING SYSTEMS AND LANGUAGES

(6)
ARE ADDITIONAL CARDS/PERIPHERALS REQUIRED? IS MEMORY ADEQUATE? MORE DRIVES NEEDED? THIRD PARTY SOFTWARE?

(7)
MAINTENANCE COST FOR FIRST YEAR. LOCAL SERVICE AND TO WHAT EXTENT? HOW MUCH LOCAL EXPERIENCE?

(8)
SINGLE USER VS MULTI-USER SYSTEMS. RESPONSE TIME? EXPANSION COSTS? MTBF FOR COMPONENTS AND IMPLICATIONS FOR INSTRUCTIONAL PROGRAMS.

(9)
5 YEAR PROJECTION FOR MAINTENANCE COSTS. THIS IS A GUESS, BUT DO IT ANYWAY! IS IT A STANDARD OFF-THE-SHELF SYSTEM OR A "HOBBY-TYPE" SYSTEM?

(10)
TYPE OF MICROPROCESSOR, ONE OR TWO? 8 VS 16 BITS. BYTES FREE OR WORK SPACE PER STUDENT. QUALITY OF MANUALS AND DOCUMENTATION. EXTENT OF VENDOR-INDEPENDENT LITERATURE. EXTENT OF USER GROUPS.

(11)
STAFF TRAINING REQUIRED FOR USE. TIME REQUIRED FOR OPERATION AND MAINTENANCE BY TEACHERS.
HUMAN FACTORS: KEYBOARDS, CRTS, ETC.

(12)
TOTAL COST: HARDWARE, SOFTWARE, MAINTENANCE, TRAINING, TEACHER TIME, SCHOOL SYSTEM SUPPORT, ESTIMATED LIFE CYCLE COST.

should, therefore, be structured to focus on what will be taught, why students ought to use computers, and how computer use will be linked to the curriculum. A structured interview is a powerful tool for establishing need.

STEP 2: SPECIFIC USE Given a curricular basis for using computers, this step focuses on specific applications. What software and/or languages are required? Again, effort is made to keep conversation away from brand name hardware systems. Educators, like most professionals, have a tendency to want to use machines they already know how to operate. This situation sometimes causes educators to become unpaid "sales people" for particular manufacturers. Institutional computing is different from hobby-type computing and as good as some machines are for home use, they may not be appropriate for classrooms. Thus, emphasis is placed on the characteristics of the required software.

STEP 3: OPERATING SYSTEMS Although somewhat technical for most educators, the operating system of a microcomputer has much to do with how useful the hardware will be in the future. With knowledge of instructional purposes and a list of specific software requirements, it is now possible to narrow the number of machines that will do the job. It is also possible to look at the availability of new software in terms of market trends and to consider the fact that the equipment will have to serve a variety of uses. Here the task is to list appropriate operating systems and to ask questions about software transportability.

STEP 4: THE MARKET PLACE This part of the process involves actively seeking out machines that meet the requirements described in steps 1-3. Vendors should be contacted and the literature of professional and technical journals should be searched. Attending meetings and trade shows is also helpful. This phase is purely informational and is not a Request for Proposal (RFP). Keeping up with hardware trends is an ongoing activity.

STEP 5: FLEXIBILITY This step is optional but important. Other things being more or less equal, a machine providing for the use of additional operating systems or languages is probably a better investment. In fact, one measure of design and engineering quality is the extent to which this is possible. This does not mean that single purpose systems ought to be excluded. However, since hardware selection will follow identification, this kind of information will be useful.

STEP 6: ADDITIONAL REQUIREMENTS Adding to a basic machine is always expensive. On some computer systems, the disk drives and controller cards cost more than the CPU and keyboard. Moreover, third party sources for firmware and extra boards and interface cards divide responsibility for proper operation. Steps 2 and 3 established specifications for instructional use and, in most cases, users will be better off by dealing with one vendor. And then there is the question of continued availability of parts. In general, the more that needs to be added, the more likely there will be problems.

STEP 7: MAINTENANCE The cost of maintenance for hardware must be considered when identifying equipment. This is an area where much improvement is needed. Some questions to ask are as follows: Is on site repair provided? Is it possible to train school system staff to do some kinds of maintenance? Did design of the hardware include ease of maintenance as a quality factor? Does the manufacturer provide timely and appropriate training for local vendors? How many different local business people provide repair service? These are a few of the direct questions that must be asked of potential vendors, and it is a good idea to ask in writing.

STEP 8: SINGLE VS. MULTI-USER SYSTEMS In this area two things are important. First, is any computing power lost by using a system consisting of satellite stations and a controller? Some relatively low cost systems look good until tasks such as file manipulation are analyzed. Again, consider machine capability in the light of instructional purpose. Second, the more complex a system is, the more likely there will be down time. If all student work stations depend on one machine, then the risk is high. Reliability is important.

STEP 9: LIFE CYCLE COSTS Use information from previous steps to look at life cycle maintenance costs. Some vendors charge as much as 20% of original hardware cost per year for service. Here the point is that in five years the system is "purchased" again!

STEP 10: REALITY TESTING If all looks good up to this point, then it's time for some serious testing. Obtain hardware systems for evaluation. Most vendors will loan machines for this purpose. Use the machine to do what will be expected of students. How readable is the documentation? Is 64K of RAM really 64K bytes free? How much literature is available about the use of the equipment? Ask many questions and keep records of responses.

STEP 11: TRAINING AND HUMAN FACTORS Teacher training is one of the hidden costs in implementing a computing program for students. Another factor of operational significance is the time required of teachers to maintain and use computers on a daily basis. The quality of keyboards and CRTs will also have an impact on how teachers and students view the proposed use of the equipment. It is not, for example, appropriate to teach word processing on a membrane keyboard. In the end, human factors will have much to do with machine use and overall program effectiveness.

STEP 12: THE BOTTOM LINE At this point, the data and observations regarding possible hardware systems are put in tabular form and the advantages, disadvantages, and costs are compared. The process may also be used to generate criteria for the selection of hardware when considering bids.

The process described in this paper is not structured analysis in the formal sense. Authors such as Tom DeMarco and Victor Weinberg have defined this discipline in a more rigorous manner. There is, however, common ground: the use of graphics; building logical specifications based on needs; the use of structured interviews; and establishing objectives with cost data related to purpose.

"THE COMPUTER AS A CLASSROOM TOOL" - "A STATEWIDE INSERVICE PROJECT"

Need for teachers and other educators to be trained in technology has rapidly moved to the forefront in the last two years. WEA passed policy in April of 1982 encouraging development of a training program. A review of options available on a statewide basis showed some training available in programming on specific types of microcomputers and a variety of special interest or general awareness workshops available. No comprehensive program for training teachers regarding microcomputers relevant to classroom uses and needs was readily available from either a state or national level. The following program or plan is an attempt to meet that need -- experience to date indicates that the need is even bigger than we can meet with what we perceive to be a fairly ambitious training plan.

It may be important to recognize here some of the traditional complicators to an undertaking of this size. All of these have been dealt with with a degree of success and we are sure a continuing positive outlook will overcome further barriers.

1. Finding instructors with appropriate credentials who can relate to other teachers. (There are 15 in this model).
2. Funding time for joint curriculum development is expensive.
3. Reluctance of universities to accept fifteen adjuncts on a statewide basis.
4. Publication of class materials and instructor manuals and purchase of support software, filmstrips, etc. are costly and time consuming.
5. Program evaluation must be elaborate to justify "doubters".
6. Flexibility in program design must meet specific needs and not just be a "programmed" course.
7. Many persons must be consulted to avoid role conflicts.
8. Commitment and follow-through is an essential element.

Why the Washington Education Association? WEA is currently coordinating a staff development training cadre of over seventy-five instructors. Topics such as Stress, Multicultural Education, Discipline and Child Abuse are a part of the WEA program. WEA's commitment to quality staff development with a classroom focus is well recognized. Our members asked us to do something!

How can a statewide program be developed and implemented in six months? First, a group of five teachers committed to utilizing computers in schools were brought together for two days to determine "What Does A Classroom Teacher Need To Know". They were asked to develop a format for writing a curriculum and for field delivery of the content once it was written. Twenty-five concepts were developed that a teacher should "know". It was determined that concepts should be integrated into learning modules to provide flexibility in meeting specific needs in that not every concept would have to be covered in every workshop. Also, there should be flexibility for non-credit or credit and that credit should be variable according to the number of contact hours. The format for development of the curriculum was to look at four components:

1. Concept or learning objective
2. Teaching activities to teach the objective
3. Resources necessary for instructor
4. Evaluation of instruction and resources

With these ideas in mind, all fifteen Washington State Institutions were contacted to invite them to participate in a cooperative project. Fifteen teachers who were "into computers" were selected to be a writing team. They represented all regions of the state, all grade levels, librarians, business education, etc. Their diversity rapidly became their strength. This team came together as strangers for six days in August and utilized the previously described model to develop a curriculum and identify needed resources. A manual was prepared in draft form which was reviewed in a three-day meeting in October. Two universities had participated in all these activities, while others had limited involvement.

The following is a brief content and delivery description.

MODULE I - KNOWLEDGE OF COMPUTERS
This beginning module introduces the fundamental concept of the computer; how the computer works; the major terminology needed to understand and converse with others about the computer; and provides initial "hands-on" experience with the computer.

MODULE II - CONTROLLING THE COMPUTER
Participants are introduced to "computer language," flow charting and problem solving on the computer. Students will gain skills in controlling the computer through using BASIC language.

MODULE III - USING COMPUTERS IN TEACHING
Participants will evaluate computer software and hardware for quality and learn how to match needs with appropriate teaching strategies. Participants will review appropriate learning styles and possible computer applications in teaching.

MODULE IV - EDUCATIONAL ISSUES
Participants investigate educational issues and implications of computers. They explore developing curriculum for the classroom; management of the computer within schools and review framework for solving computer-based problems which confront teachers.

MODULE V - THE COMPUTERS IN SOCIETY
Participants synthesize the historical aspects of computer evolution with the current issues relating to expanding computer use. They trace the evolution of the modern computer, including recent breathtaking developments, with far-reaching implications for the future. Further, they examine the controversial issues of the use of computers in society.

Each module is at a different length as they were designed to meet contact time not hour restrictions. They range from 4-6 horus in Module IV to 10-12 hours for Module II. Any combination with 10 contact hours can equal one university credit with these being a possibility of up to 40 hours of training for a total of 4 quarter hour credits. Non-credit workshops can also be conducted by individual module.

EXTENSIVE BIBLIOGRAPHY
PROVIDED BY MODULE

1-4 Credits

PLU Educ 501	UW Educ 494
UPS Educ 508	SPU Educ 5162

Total Credits identified by class
COST: $44 per Credit WEA Members
$54 per Credit Non-Members
$34 per 10 hours Non-Credit WEA Members
$44 per 10 hours Non-Credit Non-Members
Maximum 3 students per computer-
Maximum: 24 per class

The class has been initiated at over thirty sites in the state in the last four months. Evaluations evaluate every objective, handout and activity. A student evaluates on a five point rating scale. Some sample items of 115 total are:

"Students will be able to list the major components of a computer."

1. This objective was achieved — H 1 2 3 4 5 L
2. This objective was relevant to the module — H 1 2 3 4 5 L

The following activities and materials contributed to meeting these objectives.

3. Handout - Vocabulary Worksheet H L 1 2 3 4 5

4. Activity - The Human Computer H L 1 2 3 4 5

5. The instructor was well prepared for class H L 1 2 3 4 5

Demographic data is collected on work experience, education position, credit and sex. Every instructor receives regular computer printouts after teaching each module which compares their ranking on each item with the average of all other instructors. Evaluation results look exceptional at this point. Data will be available in May.

An evaluation committee with four university representatives and four WEA representatives now oversee the program content and the supervision of instructors. All fifteen K-12 teacher authors are adjuncts for the four universities listed.

The number of requests from school districts, the positive relationship with other computer groups and vendors and the cooperation provided by the State Department of Education are appreciated.

WEA has underwritten the total startup cost which was approximately $20,000. WEA is copywriting the course content.

For additional information contact Bob Pickles (206-941-6700) or write to the WEA, 33434 - 8th Avenue South, Federal Way, WA 98003.

Plans are now being developed to add additional training modules.

Possible additional training modules anticipated in Logo/Pilot, Handicapped and Computers, Migrant Education, Bilingual Education and perhaps some specific subject-matter areas.

Many of the early concerns about availability of hardware and software appear to have been overcome. Demonstration disc using public domain software have been prepared for Apple, Atari, Commodore and TRS 80. All instruction has been prepared to be offered on the four modules of micro computers. We have been able to locate training sites where we can meet at least a 1-3 ratio of computers to students.

Students are exposed to the "computer world" from the beginning of any workshop or course by filling out WEA's own computerized registration form. As implied in the beginning, this "team" of 13 plus people have accomplished a lot in less than one year.

CAI COURSEWARE DEVELOPMENT THROUGH INTERACTIVE PROMPTING

Gerald N. Pitts
Maurice L. Eggen

Trinity University
715 Stadium Drive
San Antonio, Texas 78284
(512) 736-7480

ABSTRACT

A metasoftware system called CAPS (Course Authoring Prompting System) was the result of a threefold research effort undertaken in the summer of 1982. CAPS provides a capability for CAI Courseware development in any subject area. The potential author need not be versed in computer languages, have any computer background or any previous CAI experience.

Through a structured approach, CAPS provides information concerning a) CAI methodology and background, b) development of course and behavioral objectives and, c) course structuring. This is readily accomplished through friendly interactive prompting by CAPS. CAPS's final product is a customized course that contains not only the course proper but also a student teacher communication interface as well as an automatic grading and statistical package.

Definition of CAI

Computer assisted instruction (CAI) is the term used to describe the process of teaching with the aid of a computer. It is the procedure whereby written and visual material is presented in a logical manner to a student by a computer.

A one-student-to-one-teacher relationship has long been considered the best learning situation. In the United States, the attempt to provide an education for everyone has all but eliminated this possibility. However, the use of CAI is a step toward more individualized instruction since the student is allowed to proceed at his own pace and with immediate feedback concerning his grasp of topics as they are presented.[1[

A "teaching machine" is a product of programmed instruction. The key to programmed instruction is the breaking down of concepts to be learned, into small, logically linked pieces called frames. There are three main ideas associated with the programmed instruction concept developed by a Harvard psychologist, B. F. Skinner. The three concepts are as follows: an active response by the student, an immediate confirmation or denial of that response, and the concept of a linear presentation of frames, either introducing a new idea or reinforcing an old one.[2] Skinner found that confirmation must occur as soon as possible.[3] The confirmation in a CAI lesson is the advancement to the next frame. The denial occurs if a question is answered incorrectly. The incorrect answer elicits either a corrected response or remedial instruction. Today, most CAI is still programmed instruction on a computer.[4]

The simplest mode of CAI, in which the programmed instruction concept is most obvious, is referred to as drill and practice CAI. This mode of CAI does not present new material to the student. Rather, it is designed to give the student additional practice concerning instruction already presented to the student by the teacher. The computer can keep track of the questions answered correctly by the student. Also, if designed to do so, the computer can build a model of the student, so that additional lessons can be oriented toward the topics that the student's responses indicate he needs the most help in.[5]

A second mode of CAI is the tutorial mode. This mode of CAI does present new material to the student. Either one or more frames present a new concept to the student. Then, the student is asked to respond to one or more questions to determine if he has understood the material just presented. If his responses indicate partial understanding, he may receive additional information on the same topic. However, if his responses indicate no understanding of the recently presented concept, he may need to repeat some of the earlier information. The tutorial mode of CAI is designed to be a model of a tutor who responds to the individual student's progress.[6]

A third mode of CAI is the dialogue-inquiry mode. This mode requires more sophistication in programming skill and in available software. Dialogue-inquiry is the method by which the student and the computer "converse" concerning the topic under discussion. It has been referred to as the "Socratic system" because it is similar to the method of learning espoused by Socrates in ancient Athens. In this mode, a problem is presented by the computer. The student and the computer can then discuss the problem or concept.[7] There is a restriction upon the vocabulary which the student may use. However, the vocabulary is usually rather large. Implementation of this mode of CAI requires a large data storage facility and natural language processing capability. The success of a CAI program written, using the dialogue-inquiry mode, is largely dependent upon the ability of the author to anticipate the questions and responses a student may make.

Software and Courseware Development

Software refers to the authoring programs which are available for the programmer, or educator, who wishes to develop a CAI lesson. Courseware is the product or lesson produced. The courseware is, by definition, interactive with the student's responses controlling the flow through the lesson. The authoring program may be interactive or batch oriented.

Growing out of the programmed instruction concept, CAI courseware is designed to be frame oriented. The student is asked a question. If the response is the anticipated one, then the student is told so and proceeds to the next unit. If the student does not respond in the manner anticipated, then he may be given additional chances in the drill and practice mode (depending upon the author's program) and in the tutorial mode, the student will probably receive some additional explanation. These are the general techniques but variations are possible, depending upon the inclination of the author.

The development of courseware has been as varied as the differences in the authors themselves and their approaches to instruction and education. The development of software has been a very machine dependent process. Unlike general use programming languages, such as COBOL and FORTRAN, there has been no real standardization achieved. Thus, IBM has a program product called COURSEWRITER, while UNIVAC's product is called ASET (Author System for Education and Training). PLATO, which is developed through the authoring language TUTOR, is implemented on CDC (Control Data Corporation) equipment. Digital Equipment Corporation (DEC) supports a CAI language called DECAL (Digital Equipment Corporation Authoring Language). This list could continue. This proliferation of CAI languages is bewildering to the potential CAI author. Because CAI is used primarily by non-computer scientists, the requirement of learning a CAI authoring language is prohibitive. A search of literature reveals no active research in the area of CAI authoring language standardization or author prompting systems as this research effort accomplished.

Enter Caps

CAPS, Course Authoring Prompting System, is a system of programs designed specifically for the Apple Microcomputer to allow the potential author to create a course to be administered by the computer. CAPS is a new method that allows the creation of a course without any knowledge of computer systems or programming languages. Prompting provides the technique of obtaining the necessary information for course creation. This prompting is done in English just as if questions were being asked by another human being. Each course is customized to meet individual educational objectives. However the text of the created course is unique and is structured under the guidelines of CAPS. Each step in the course development process is accompanied by extensive explanation and specific examples in order to dispel any feelings of inadequacy concerning educational objectives or computer assisted course development. CAPS provides information on Courseware Methodology. This is where potential authors learn the general procedure for creating a course. CAPS reviews educational concepts. This may be skipped by knowledgable educators as can the following section which gives a description of CAI benefits. The final section contains a comprehensive CAI example. This gives authors an idea of what their finished product will look like.

Course Methodology

CAPS presents instructions on how the individual courses are created. CAPS prompts the author to think carefully about the objectives of the course and prompts the author to enter the course objectives at the correct time. CAPS also reminds the author of his course objectives at appropriate times, and stores the course objectives for presentation to the student.

CAPS provides specific instructions on how the course will be presented to the student but only requests the author to break his course into lessons. CAPS itself will frame the lessons into screen size segments for presentation to the student.

Educational Objectives

A sample of text included in CAPS designed to assist the author concerning behavorial and educational objectives is as follows:

Since the early 50's educators have utilized Taxonomy of Educational Objectives which include behavorial objectives for lesson creation. Dr. Benjamin Bloom's cognitive taxonomy has evolved as the primary criteria for development of behavorial objectives. Bloom's taxonomy as well as other taxonomies were covered previously by CAPS which should have provided you with the conceptual background needed to utilize the behavorial approach to lesson preparation. For your review CAPS presents again Bloom's Cognitive Taxonomy. Please notice the six general categories, the first three of which provide the primary basis for most lesson preparation.

It is up to you, the author, to find the proper category from Bloom's Taxonomy. However the proper category is not critical, and if you cannot ascertain the proper category, CAPS will utilize knowledge of Specific Facts as the primary Behavorial Objective. CAPS will now again show you Bloom's Taxonomy. Please take special notice of the three digit numbers prefixing the categories. Choose the category that best fits your lesson objective. Record this three digit number on a piece of paper, on your hand, or somewhere, so that when CAPS requests your category you will have it available to input. If you can not find a category record the 700 for input to CAPS as an indicator that you want CAPS to utilize the default behavorial objective.

The primary need for the behavorial objective is in the creation of the interrogative statements (questions), CAPS will not develop the questions for you but will provide a reminder of your behavorial objective and a list of appropriate verbs that should be utilized in the question development. The behavorial objective you choose need not influence your entire course. Each time you begin a new unit you will be given the opportunity to again choose the appropriate behavorial objective.

Research Methodology

CAPS is the result of a threefold research effort undertaken by the authors in the summer of 1982.

Phase I.

An extensive search for new research methods in CAI through University Libraries, as well as personal communication with other CAI researchers, was launched. Special emphasis was placed upon review and critique of the current CAI authoring languages, in order to build an extensive collection of language similarities, standards, syntax compatibility, as well as differences. This collection was the foundation for the design and development of a standardized CAI portfolio that was the thrust of Phase II.

Phase II.

The design portfolio developed in Phase I was utilized to create standard guidelines for (a) behaviorial goal development, (b) text goal development, (c) course general objectives, (d) lesson general objectives, (e) testing objectives and (f) author training.

The development of the guidelines required extensive background and experience in CAI laboratory courseware planning and development. Knowledge of current CAI languages, in addition to knowledge of educational taxomonies, was critical at this stage. The results of this design phase were general guidelines for the development of the authoring prompting system to be constructed in Phase III. Note that very little research has been launched in the area of prompting systems even though there was a great expression of need.

Phase III.

The initial thrust of Phase III was to incorporate the knowledge gained in Phases I and II into workable programmed modules for authorship.

The development of the modules required exhaustive computer system manipulation, software patching, and machine interface, in order to establish a comprehensive authoring system that provided a "turnkey" operation for potential non-computer scientist CAI authors.

In order to incorporate the complicated logic flow described in the system flowchart of Phase II, a series of software utility programs had to be developed. These included textfile creators, readers, blockers for display, program segmentation, chaining and text editing.

Summary, Conclusions, and Future Research

The final product of this research project (CAPS) is a viable software package that can be used by educators to develop computer aided courses with any computer knowledge. The package was tested and given favorable ratings by several educational experts. Only time and testing will tell if the system is extensive enough to handle an evergrowing demand for CAI courseware by non-computer scientists.

Additional capability developmental areas surfaced during the construction of CAPS, but were beyond the scope of development because of the time period. The major areas encountered were (a) simulation and modeling capability, (b) unit review capability, (c) graphics capability, (d) dialogue inquiry capability, and (e) calculator mode capability.

In summary, the CAPS prompting system is a tremendous achievement in Computer Aided Instruction. It provides a capability for the non-computer scientist that has been sorely needed. This system provides a broad base for future CAI research. The unique concept of development of the system on a popular microcomputer provides great potential for widespread application, because of the popularity and the relative cost of microcomputers. It is these author's hope that CAPS reaches it potential in CAI application.

A PROGRAMMING LANGUAGE FOR ELEMENTARY STUDENTS

Thomas J. Prosen
Instructional Coordinator
Minnesota Educational Computing Consortium
2520 Broadway Drive
St. Paul, Minnesota 55113

ABSTRACT
With the frontier of computers in education upon us, the microcomputer is reaching all areas of our educational systems. As students use the microcomputer with drill and practice, tutorial, simulation, and problem solving types of programs in their various classrooms, they begin to wonder how they can "make the computer do what they want it to do".

PILOT, which stands for Programming, Inquiry, Learning Or Teaching, is a programming language which can achieve this goal for the non-programmer. ATARI PILOT is ATARI's Version of the PILOT Language to be used on the ATARI 400 or the ATARI 800 computer. Although the PILOT Language has been implemented on other computers such as the APPLE II and can be used as a development language, this article will focus on the potential of ATARI PILOT as a tool to help teach programming to elementary students.

Although BASIC is the computer language most commonly taught in schools, some of the features of PILOT, especially the ATARI version, make it an attractive alternative for introduction computer use to children. Traditional types of programming may not interest many young students. But incorporation of sound and graphics into a program, a specialty of ATARI PILOT, makes programming a more motivational activity.

ATARI PILOT uses what is called Turtle Graphics (referring to early experiments in teaching programming to children which utilized a mechanical "robot" that resembles a turtle). Turtle Graphics enables a young student to draw pictures on the screen much easier than in regular BASIC. The needed commands are very easy to understand and implement.

Another item in ATARI PILOT which will immediately eliminate some frustration is that students need not worry about line numbers when writing programs if they move into the AUTO MODE. Upon leaving AUTO MODE, the program, when listed out, will look as though the student had written it with line numbers included.

The use of sound has also been made easy enough to allow young students to write computer programs which will produce music on the computer through the speaker on the television.

In the following examples, reasons why ATARI PILOT is easier to use for most students than ATARI BASIC will be demonstrated. To begin with, most young students will become eager to program if we introduce them to the "world of turtle graphics" to enable them to draw a picture. With a few simple commands the student could draw a picture on the screen. Shown in the first example is the program needed to draw a simple square written first in ATARI PILOT then in ATARI BASIC. The PILOT program will not have any line numbers since the student would not need to use them unless desired.

TM Apple II as used in the paper is a trademark of the APPLE Computer, Inc.

R ATARI and ATARI PILOT as used in the paper are registed trademarks of ATARI, Inc.

ATARI PILOT

As we look at the PILOT program it is obvious that the student can actually imagine himself as though he were moving around the room in the direction that the imaginary turtle is being moved on the screen. In this way it is as though the student is in his own self-contained environment. Since young students refer to locations in terms of where they themselves are located, the transition to a turtle is quite simple. Explanatory notes are added in the parentheses.

```
GR:DRAW 30 (30 units vertical)
GR:TURN 90 (turns a right
            angle clockwise)
PA:100  (creates a pause)
GR:DRAW 30  (30 units horizontal)
GR:TURN 90
PA:100
GR:DRAW 30
GR:TURN 90
PA:100
GR:DRAW 30
GR:TURN 90
E:
```

This is a very simple example but let's look at the same program in ATARI BASIC.

ATARI BASIC

Notice as the program begins that besides having to know about different graphics modes and color, the student must have an understanding of the Cartesian coordinate system to know in which place to plot the next point. Also notice that instead of a simple pause command the student must understand FOR-NEXT loops and possibly subroutines. By the time we finish explaining and demonstrating all of that information many students would probably lose interest.

```
10 GRAPHICS 6 (graphic mode)
20 COLOR 3  (sets color)
30 PLOT 80,45 (first point)
40 DRAWTO 80,15
50 GOSUB 500  (pause method)
60 DRAWTO 110,15
70 GOSUB 500
80 DRAWTO 110,45
90 GOSUB 500
100 DRAWTO 80,45
110 GOSUB 500
120 GOTO 600
500 FOR X= 1 TO 500
510 NEXT X
520 RETURN
600 E:
```

Since we have a square on the screen we may want to add some statements to fill it in with a solid color. This time, however, we will first observe how this is done in BASIC and then see how comparatively simple it is in PILOT.

ATARI BASIC (additional statements needed to be placed in the previous BASIC program)

```
110 POSITION 80,15  (to fill the square you
                     would need to have
120 GOSUB 500        advanced know-
130 POKE 765,3       ledge of BASIC)

140 X10 18,#6,0,0,"S"
150 GOSUB 500
160 GOTO 600
```

ATARI PILOT

```
GR:FILL 30
```

We need to include only one additional statement to enable us to fill the square in ATARI PILOT. This statement would come directly before the E: statement in this case. It is not only much simpler but it also makes much more sense that to fill something you can use the word FILL.

ATARI PILOT also has a repeat statement which makes life easier since it requires less typing when the same commands are performed over and over again. The filled in would take on the form as follows:

```
GR: 4(DRAW 30; TURN 90)
GR: FILL 30
E:
```

Using the commands we've seen thus far and a small amount of experimentation, a student will soon be able to come up with a circle. This is a very difficult task if one is programming in ATARI BASIC at the elementary level. With ATARI PILOT it would be performed as follows:

ATARI PILOT

```
GR: 120(DRAW 1;TURN 3)
E:
```

To draw a circle in ATARI BASIC is no simple task. We need to use the equation of a circle to do the best job. So let's take the equation $X^2+Y^2 = 25$. Solving for Y we get $Y=SQR(225-X^2)$ and this can be positive or negative. So our knowledge of mathematics must be much more advanced than the knowledge necessary to draw a circle in ATARI PILOT. The BASIC program would be as follows:

```
10 GRAPHICS 6
20 COLOR 3
30 FOR X = -15 TO 15 STEP .5
40 Y=SQR(225-X^2)
50 PLOT 20+X,30+Y
60 PLOT 20-X,30-Y
70 NEXT X
80 END
```

Another very motivating operation available with ATARI PILOT is a simple method of playing music. With a single command (SO:), one to four notes can be played at one time. Each number following the SO: command represents a specific note taken from a chart of notes found in the PILOT reference manual. Used with the PAUSE(PA:) command the note or notes may be held for a pre-determined length of time. Four chords each held for an equal period of time are produced by the following ATARI PILOT example:

```
SO: 1,5,8,13      C MAJOR (C,E,G,C)
PA: 50
SO: 5,8,13,17      1ST INVERSION
                     (E,G,C,E)
PA: 50
SO: 8,13,17,20      2ND INVERSION
                     (G,C,E,G)
PA: 50
SO: 13,17,20,25      C MAJOR 1 OCTAVE
                      HIGHER (C,E,G,C)
PA: 50
```

The pauses may be modified to change the rhythm.

The motivation one receives from programming with sound and graphics is important, but if we look at a simple BASIC program which merely asks two questions and uses the answers in a sentence, we will still be able to observe how much less technical knowledge is needed to program using PILOT.

ATARI BASIC

```
10 DIM NAME$(20)
20 PRINT "WHAT'S YOUR NAME?"
30 INPUT NAME$
40 PRINT "HOW OLD IS ";NAME$; TODAY?"
50 INPUT X
60 PRINT NAME$; "IS ";X; "YEARS OLD."
70 END
```

As you can see the student would need to know about semicolon, quotation marks and spacing which cause problems for beginners. The same program would look like this using PILOT.

ATARI PILOT

```
T:WHAT'S YOUR NAME?
A:$NAME
T:HOW OLD IS $NAME TODAY?
A:#X
T:$NAME IS #X YEARS OLD.
E:
```

There are many more PILOT commands which enable a student to produce more advanced programs. The ones discussed here should enable teachers to realize the usefulness of ATARI PILOT at the elementary school level and perhaps at the junior or senior high level as well. As with any computer language, PILOT will not meet the needs of everyone. It does seem to provide an improved method of introducing the concept of computer programming to children in a way which enables them to quickly take advantage of the personal computer.

PREPARING TEACHERS FOR A COMPUTER SCHOOL

Dr. Raymond Ramquist
James Madison University
Harrisonburg, VA 22807

ABSTRACT

This presentation reports on efforts to review literature in regard to what higher education is doing to prepare future teachers to become computer literate. It then presents a model, as developed at James Madison University, for preparing teacher education majors to become computer literate. As part of the model, efforts are outlined for preparing the university faculty in computer use, and a listing of computer competencies is presented that all teacher education students will be expected to demonstrate.

THE PLAN

The School of Education and Human Services at James Madison University was in need of a plan to integrate some form of computer awareness into the training of its undergraduate students. The School was further faced with demands from school divisions to provide courses and inservice experiences for teachers in the public schools. The second problem was addressed first and had an outcome to help with the first problem.

In the summer of 1982 an inservice workshop was offered to twenty faculty from the School of Education and Human Services. The hoped outcome of this was to begin the training and identification of faculty from the various education departments who could take leadership roles in using computers in their own teaching and in providing a core of faculty that could teach inservice courses off campus. Preceeding this effort was the development of a one to three credit course outline for a workshop that could be delivered to school divisions. In the 1982-83 academic year, seven faculty were identified as teachers who would teach off campus. During that year a total of thirteen courses ranging from one to three credits were offered to eight different school divisions. Considering that the previous year no computer courses were offered to the public schools, this was a significant increase.

The second problem was not as easy to solve. The present teacher training programs were already heavy with required courses and trying to add another required course would meet heavy opposition from the faculty. Three authors were cited as a basis for why computer literacy should even be considered as part of the teacher training curriculum. Molnar stated that society is on the brink of moving from an industrial society to an information society. This was not anything new but was the beginning of the argument. Dr. John Beck saw four main objectives for having people and teachers computer literate:

to reduce the bewilderment about computers in the minds of individuals and to promote a balanced view of the computer's role in society.

to enable people to use and influence the design of computer based social services with confidence.

to enable people to work in a computer environment without major difficulties.

to develop informed opinions to influence decisions regarding computer applications which have political and educational implications.

In a paper presented by Dr. Richard Diem, he cited a study done in 1980 that found out of 227 student teachers 94.4 percent felt unqualified to deal with computer literacy topics. Finally and perhaps the most sound basis, as to the why of computer literacy for teachers, was stated by Dr. Gerald Gleason: "based on the assumption that a computer literate society has its roots in the schools, it follows that schools must have teachers who themselves are knowledgeable about computer applications and potential."

These authors gave a basis for the why of computer literacy, but there still remained the problem of how to get computer literacy to the undergraduate teacher training student. The easiest solution was to find out how other schools of education taught computer literacy and use those examples. A search of current literature provided less than fruitful examples. A bibliography search by Bibliographic Retrieval Services found only 45 citations and of those only 13 dealt with literacy for teachers and/or future teachers. These reports had few concrete examples of what was actually done to train future teachers. As part of the search, the report by Dr. Richard Diem was found to be useful as a basis for James Madison University's plan.

Dr. Diem presented three models: 1) Require a computer science course for all teacher education students. He discounted this approach. 2) Use a course in educational technology geared to computers. This approach was not possible since James Madison University didn't already have a required educational technology course. Further, Diem didn't favor this model as it tended to compartmentalize the exposure to computer technology. 3) Integrate and apply computer technology throughout the entire teacher education curriculum.

This integrate/apply approach was how James Madison University was presently providing other technology competencies into the teacher education curriculum. All students were participating in a series of learning stations whereby they learned how to operate various types of audiovisual equipment, and then apply those competiencies in methods classes. Students were also learning how to turn on the microcomputer and interact with computer programs. This was accomplished with a learning station and a disk program titled Apple Presents Apple.

Since the integrate/apply approach was the model decided upon, it now became necessary to determine the competencies that all education students would need to operationalize this model. This task was supported by the wide range of literature that already exists in the form of books and articles explaining what is taught to young people and others interested in learning how to use computers. The following is a list of competencies that all education students will be able to demonstrate by the time they graduate from James Madison University.

A. Role of computers in education. The student will be able to describe:
   1. The history of computers and their application to education.

   2. Research findings relative to the effectiveness of computer

assisted teaching.

3. Different philosophical orientations in education as they relate to computer assisted instruction.

B. Terminology.
The student will be able to:
1. Find and/or describe the Printer, Disk Drive, ROM, RAM, CPU, Bit, and Byte.

2. Use the different files on a disk: Text, Integer, Machine Language, and Basic.

3. Demonstrate changing a disk from DOS 3.2 to DOS 3.3 using the Muffin program.

4. Demonstrate what is contained on a disk catalog.

5. Describe the difference between input and output and how each is accomplished.

6. Describe the different variables a computer uses: numeric, string, integer.

7. Describe the difference between computer memory and storage capacity on a disk.

8. Describe the differences in compatability of software for various machines.

9. Describe how to protect a program from being erased.

C. Mechanical Operation.
The student will be able to:
1. Set up a microcomputer for operation.

2. Insert a disk and bring the machine into operation.

3. Load and run a program from a disk.

4 Exit from a program.

5. Interact with a program using the Return and Arrow keys.

D. Selection and use of courseware.
The student will be able to:
1. Select software using various sources and selection aids.

2. Describe the differences between various software programs such as: drill and practice, tutorial, simulation, problem solving and decision making.

3. Apply criteria of selection to software such as cost, compatability of equipment, curriculum fit, user friendly, flexibility and attainment of objective.

4. Apply different instructional strategies when using CAI software.

E. Programming.
The student will be able to:
1. Describe the various programing languages (Basic, Pilot, Logo, Aids) and how they can be used.

2. Flow chart an instructional problem and break it into its decision point parts.

3. Create a simple CAI program using the commands of PRINT, INPUT, IF THEN, GOTO, GOSUB.

4. Demonstrate the ability to create graphics on the microcomputer.

F. Specific Department Requirements. Each department will identify those additional computer competencies which are essential for graduates of each of its major undergraduate programs and provide for the development of these competencies.

The steps involved in accomplishing this approach are:

1. Every student will take a fifteen hour laboratory experience as part of an already existing introductory education course required of all teacher education majors. This will occur in the junior year and will be conducted in the Educational Media Labs through self instructional learning stations and assistance from a graduate teaching assistant.

2. Every methods course in the teacher education program will include additional instruction related to micro-computer software which is relevant to that teaching area.

3. The School of Education and Human Services will endeavor to make appropriate use of computers in accomplishing its mission so that students will have had first hand experience in observing the use of computer technology from the consumer point of view.

One task that yet needs to be accomplished to help attain the final outcome of the integrate/apply approach is to provide for raising the computer literacy level of the teaching faculty in the School of Education and Human Services. As will be recalled, this outcome has already been started with the training of twenty faculty. This will continue in the Summer of 1983 with forty more faculty, and will continue throughout the coming academic year with support from the Educational Media Laboratory to help faculty develop uses for computer software.

BIBLIOGRAPHY

Beck, John "Computer Literacy for Elementary and Secondary Teachers", November 5, 1980, ED 208 868.

Diem, Richard A. "The Role of Technology in Teacher Education; Preparation for the Twenty-First Century", February, 1982 ED 212 596.

Gleason, Gerald "Microcomputers in Education; The State of the Art" Educational Technology, March, 1981, pp 7-18.

Lockard, James "Computers Blossom at a Small School in Iowa", Instructional Innovator, September, 1980, p 25+.

Molnar, A. R. "The Next Great Crisis in American Education; Computer Literacy" AEDS Journal, December 1978, pp 11-20.

COMPETENCY BASED TRAINING OF SOFTWARE SPECIALISTS

T. J. Koehler
S. M. Raucher

ABSTRACT

Like many institutions, the Montgomery County (MD) Public Schools has found it difficult to hire software professionals with specific experience in our complex systems. The majority of new hires, who possess previous DP experience, require significant on-the-job training before they can be fully productive. As such, we have implemented a well-defined and carefully structured training program for in-house staff.

The training program at MCPS combines features of competency based training, internship experience, and mentor/trainee interactions, all within a carefully defined and individualized program for each trainee. This paper provides a detailed examination of each of these features. It provides the reader with specific data on skills/competencies, together with the training method and evaluation scheme used by MCPS. Competency areas include: COBOL Programming, Use of Utility Programs, Library Functions, On-Line Programming Tools, Report Generator Programming, Database Programming, Documentation, Implications of Internal Computer Functions, Program Testing, Structured Analysis and Standards. The paper ends with a generic discussion of the results observed over the four years that the training program has been in existence.

The Montgomery County Public Schools Model - The Environment

Montgomery County Public Schools (MCPS) is a large suburban school district adjacent to Washington, D.C. MCPS has long been involved in the application of computer technology to support school administration, business management, and instruction. In carrying out its various data processing, management information, and computer technology missions, MCPS employs over one hundred data processing employees, sixty of whom are "software specialists" (programmers, analyst/programmers, systems programmers, analysts, etc.).

Initial entry into the MCPS software staff is primarily at the most junior level, since, for higher level positions, promotion from within is the general policy. Thus, MCPS typically hires between three to seven junior staff members each year to cover attrition and program growth.

Experience has shown that in a software environment as complex as MCPS, (use of structured techniques, data base, procedural and non-procedural languages, application generators, information centers, etc.) the lost time and cost of retraining a typical "experienced" programmer (2 to 3 years experience) was essentially the same as for hiring and training a raw trainee programmer. College, university, and technical school graduates, with or without experience presented the same picture. This is illustrated by the fact that during the past four years MCPS has not had one individual programmer applicant (out of hundreds) who met the basic qualifications to function effectively in the MCPS environment without extensive training or retraining. MCPS has also had a far higher retention rate for persons trained internally in constrast to the general industry experience of very poor retention rates.

Experience has shown that in a software environment as complex as MCPS, (use of structured techniques, data base, procedural and non-procedural languages, application generators, information centers, etc.) the lost time and cost of retraining a typical "experienced" programmer (2 to 3 years experience) was essentially the same as for hiring and training a raw trainee programmer. College, university, and technical school graduates, with or without experience presented the same picture. This is illustrated by the fact that during the past four years MCPS has not had one individual programmer applicant (out of hundreds) who met the basic qualifications to function effectively in the MCPS environment without extensive training or retraining. MCPS has also had a far higher retention rate for persons trained internally in constrast to the general industry experience of very poor retention rates.

Given the desirability of training entry level software specialists/programmers, MCPS has developed and refined over a four-year period a competency-based training program for entry level personnel. The general concepts behind this program are:

°Competency Based--A functional analysis of those competencies needed to function as a software specialist/programmer were isolated and described. Measures of competency were identified and the means of acquiring these competencies were developed. This yielded a three-part matrix for each competency. A simplified example is:

Competency	Measure	Learning Method
Use COBOL Report Writer effectively	Report program coded using all features of Report Writer	Community College Advanced COBOL Class--OJT working with an experienced programmer(Mentor)

°Intern Concept - The MCPS program combines formal education with an intern program in order to develop "real world" competencies. Each trainee is assigned for three to six month periods to the systems maintenance and enhancement function and for a similar period to the new applications development function. Additional short-term assignments are served in the software support, and user liaison/information center functions.

°Mentor System - Each trainee is assigned an experienced software specialist-analyst/programmer as a mentor. The mentor role is one of helping, facilitating, tutoring, and guiding. Mentors are not responsible for training, per se. They are part of a support system. The time Mentors spend with trainees is considered as task time, and they are held accountable for providing help and assistance. Only persons with past performance in the role, or a known aptitude, are assigned as mentors.

°Individualized Training - Trainees come into the program with widely varying competencies. Each trainee is evaluated initially and every three months thereafter on where he or she stands on each competency. Where the trainee is fully competent and can demonstrate skills, no further training is required. Where skills are lacking, training is provided. While all trainees share in common experiences, their intensive training is geared to individual needs only.

STRUCTURE OF MCPS DATA PROCESSING INTERN PROGRAM

The program is directed by the development division director, and a trainee coordinator designated by the director coordinates the program. The director defines achievement standards, approves training programs, reviews performance and makes the final determination on achievement of objectives.

Interns are normally allowed no less than approximately 20 to 25 percent of work time for direct training activities (class attendance, video courses, etc.). Class attendance within the direct training allotment is considered as regular work assignment. Interns are also to be encouraged to use "own time" for training, and may accelerate their progress through after-hours self-training.

Employees in this program are ordinarily classified as "programmer trainees" and remain as trainees from 9 to 18 months depending on the individuals. Trainees will serve a minimum of nine months in the training program. Compensation is determined by the Department of Personnel in accordance with the MCPS regulation covering trainees.

The competencies expected to be attained at the end of the training program have been the subject of development and refinement over a four-year period. This definition of competencies could prove of value and interest to developers of industrial and academic training programs. In addition, given the great disillusionment with software educational efforts, these competencies could serve as a useful standard against which to compare academic programs.

I. COBOL PROGRAMMING

Skill: Ability to code new programs in COBOL using internal sort, subscripting/indexing, database, file-matching (as in updating a file) and report writer in a structured manner.
Training Method: An advanced COBOL course similar to a course offered at a community college, or completion of video tape courses and/or on-the-job training with assistance from a mentor.
Competency Measure: Working and tested program samples exhibiting these skills.
Skill: Ability to modify existing programs written in structured or non-structured format.
Training Method On-the-job training - actually having to make changes to other people's programs.
Competency Measure: Working and tested program samples exhibiting new code in with old.
Skill: Ability to troubleshoot existing programs.
Training Method: On-the-job training.
Competency Measure: resolved trouble report.
Skill: Ability to coordinate, conduct, and document a walk-through.
Training Method: Participate in other programmer's walk throughs. Read Yourdon book on walk-throughs.
Competency Measure: Successful walk through.
Skill: Ability to use "code" generator(s) that may be in use.
Training Method: On-the-job training and manuals or other documentation.
Competency Measure: Samples of "Generated Code" produced by generator(s).

II. UTILITIES

Skill: Ability to understand and effectively use utilities available: (1) SORT, (2) Utilities, (3) Database Utilities
Training Method: in-house class on utilities most frequently used.
Competency Measure: successful JCL execution listings of utilities used, and/or read and use "Compilation of Utilities" which is list of the execution JCL for utilities; on-the-job use of utilities; review of the utilities manual; some on-the-job training with assistance from a mentor.

III. JCL USAGE

Skill: ability to use JCL necessary to compile programs and utilize librarian execute programs.
Training Methods: in-house class and/or video JCL classes; Librarian Procedures Package; review of the JCL Manual.

Competency Measure: demonstrate by example.

Skill: ability to develop PROC (procedures) that contain symbolics, JCL restarts, condition code usage, program parameter and DD statement KEYWORD overrides, data set concatenation (via execution JCL) to a data set in a PROCedure.

Training Methods: In-house classes and/or video classes and/or some on-the-job training with assistance from a mentor.

Competency measure: demonstrate by example.

IV. ABILITY TO UTILIZE LIBRARIAN:

Skill: complete understanding of the librarian commands and features.

Training Methods: in-house class and/or thorough study of Librarian Manuals and/or some on-the-job training with assistance from a mentor.

Competency Measure: demonstrate by example

V. ABILITY TO USE ON-LINE PROGRAMMING TOOLS

Skill: Command Language - ability to use commands and the PF key functions in effect while in command mode.

Training Methods: In-house class on online programming and/or in-house class on command coding; review online Programmers Manuals; some on-the-job training with assistance from a mentor.

Competency measures: demonstrate by example, and/or place a command program into production.

Skill: Edit Commands - ability to use EDIT commands and the PF key functions in effect while in EDIT command mode.

Competency Measure: Same as above.

Skill: demonstrate keyboard skills

Training Method: Enrollment in a typing class or prior proficiency.

Competency Measure: Typing at 30 wpm.

VI. REPORT GENERATOR PROGRAMMING

Skill: ability to code new program for detailed report generation.

Training Methods: in-house class and/or self-training manual and/or some on-the-job training with assistance from a mentor.

Competency Measure: working program examples exhibiting these skills.

Skill: ability to code new program for totals generation only.

Training Methods: in-house class and/or self-training manual and/or some on-the-job training with assistance from a mentor.

Competency Measure: working program examples exhibiting these skills.

Skill: ability to produce the database path necessary to extract given data.

Training Methods: in-house class and/or self-training manual and/or some on-the-job training with assistance from a mentor.

Competency Measure: working program examples exhibiting these skills.

Skill: ability to match two files.

Training Methods: in-house class and/or self-training manual and/or some on-the-job training with assistance from a mentor.

Competency Measure: working program examples exhibiting these skills.

Skill: complete understanding of the Report Generator commands.

Training Methods: in-house class and/or self-training manual and/or some on-the-job training with assistance from a mentor.

Competency Measure: working program examples exhibiting these skills.

VII. ON-LINE DATABASE QUERY PROGRAMMING

Skill: ability to retrieve a given data element from the database, basic understanding of commands and features, and ability to use Query to verify test results in program testing.

Training Method: in-house training class and some on-the-job training with assistance from a mentor.

Competency Measure: Working Query Program to demonstrate the retrieval capability.

VIII. DATABASE PROGRAMMING

Skill: ability to interpret a Bachman chart and basic understanding of CODASYL DB theory and concepts.
Training Method: in-house course on database.
Competency Measure: demonstrate interpretation of Bachman chart.
Skill: ability to code an efficient database program. (Thorough knowledge of all database commands.)
Training Methods: participate in walkthrough of database programs developed by journeyman programmers and/or read database materials.
Competency measure: completion of subschema request form.
Operational programs.

Skill: ability to request a subschema to satisfy the needs of a program, ability to interpret a subschema listing, ability to determine the most efficient path through the database, and the ability to understand error codes to resolve abends.
Training Method: some on-the-job training with assistance from a mentor.
Competency Measure: Conduct a program walk through.

IX. PROGRAMMING ON-LINE

Skill: ability to develop on-line program to query or update the database using on-line program generation.
Training Methods: in-house course and/or some on-the-job training with assistance from a mentor.
Competency Measure: operational program, and conduct a program walk through.

X. DOCUMENTATION

Skill: ability to prepare production run book documentation for setup and execution of production programs or job streams.
Training Methods: Standards Manual --samples and instructions found within those pages, and/or assistance from journeyman programmer, and/or look over run books that are used from day-to-day and are up-to-date.
Competency Measure: demonstrate by example, and/or completion of a work sample exhibiting these skills.
Skill: ability to describe user requirements in general and detail.
Training Method: assisting a journeyman analyst in a project.
Competency Measure: demonstrate by example of a completed work sample exhibiting these skills.
Skill: ability to design and develop user input forms.
Training Method - existing sample exhibiting these skills, and/or assistance from analyst/programmer.
Competency Measure: demonstrate by example of a completed work sample exhibiting these skills.
Skill: ability to add or modify documentation that exists on the data dictionary.
Training Methods: read the Data Dictionary Manuals, and/or in-house course, and/or Standards Manual, and/or some on-the-job training with assistance from a mentor
Competency Measure: demonstrate by example of a completed work sample exhibiting these skills.
Skill: ability to understand data flow diagrams and to produce lower level diagrams.
Training Methods: video courses, and/or attend data-flow walk throughs, and/or some on-the-job training with assistance from a mentor
Competency Measure: demonstrate by example of a completed work sample exhibiting these skills.
Skill: ability to write in a clear and concise manner.
Training Method: complete technical writing course.
Competency Measure: demonstrate by example of a completed work sample exhibiting these skills.

XI. FAMILIARITY WITH INTERNAL COMPUTER FUNCTIONS

Skill: ability to understand basic computer science concepts and internal machine actions and implications.
Training Methods: Intro to computer science similar to course at

community college, and/or use in-house report and technical manuals, and/or technical in-house training class

Competency Measure: Demonstrate by example.

Skill: ability to use core dumps to resolve abends.

Training Method: technical in-house training class and/or assistance from journeyman programmer, and/or assembler course

Competency Measure: demonstrate by example of use to find and solve problem.

Skill: ability to maintain a basic understanding in hardware and software advances (i.e. type, capacity, speed, characteristics, applications).

Training Method: read in-house reports, and/or technical manuals, and/or trade journals and in-house conferences, and/or assistance from operations, journeyman programmer

Competency Measure: be able to identify and discuss in general different parts of system and their operation within a logic flow.

XII. PROGRAM TESTING & IMPLEMENTATION INTO PRODUCTION

Skill: ability to thoroughly test a program exercising as many of the logic paths as is feasible.

Training Methods: on-the-job training with a project that tests all variables, and/or tests with live data after testing "made-up" data.

Competency Measure: two working samples of programs that have been running successfully about 10 times without major bugs.

Skill: ability to follow the division procedures for placing a program into the production stream.

Training Methods: this involves placing the program into production load library, and/or read procedures in Standards Manual, and/or walkthrough the process with a journeyman programmer.

Competency Measure: the above samples will also suffice.

Skill: ability to develop, implement, and document a testing plan.

Training Methods: read articles on test plans, and/or participate with a journeyman programmer in developing a test plan, and/or review existing test plans

Competency Measure: Sample test plan

XIII. STRUCTURED ANALYSIS

Skill: ability to understand the structured analysis documentation (data flow diagrams, mini specs, decision tables, etc.)

Training Methods: video series on structured analysis, and/or some on-the-job training with assistance from a mentor

Competency Measure: production program developed from structured specs.

XIV. STANDARDS

Skill: ability to comply with MCPS job standards in accomplishing the above skills and related tasks.

Training Method: each trainee will be given an up-to-date standards manual with instructions to review it and incorporate the standards in their work.

Competency Measure: familiarity with department and division standards, and internal procedures.

*NOTE: These competency descriptions are normally presented in the three-part matrix form seen on page 2 of this paper. They have been reformatted for ease of publication.

CONCLUSION

Based on the four years of experience at MCPS in training junior level programmers/software specialists, the following conclusions are advanced:

1. The general range of skills and competencies needed by a software specialist are:
   - A. COBOL as the primary general purpose language
   - B. Database interface language (DML) integrated into COBOL and other languages and thorough understanding of Database (CODASYL) Logic
   - C. Report Generator language (with database interface)

278

D. ONLINE Display (CRT) language (with database interface)

E. ONLINE program tool commands integrated with Library Program Management

F. Job Control Language including procedural development (PROCS), and execution through online commands (CLists, Execs, RPF's, etc.)

G. Documentation Skills

H. Systems Analysis and Structural Analysis and Design Skills

2. Training is best accomplished in a step-by-step method of formal training followed by practical immediate application.

3. Community resources should be used where possible for general training--community college courses in data processing fundamentals and COBOL.

4. In the modern complex systems environment, internal training programs may be the most cost effective method of acquiring and retaining staff.

5. Competency based training provides clear guidance to trainees, and a relatively precise method to schedule training and evaluate results.

6. Training should be broadbased and include documentation, analysis, structured methodologies, and systems analysis rather than merely concentrating on coding skills.

7. A policy of internal advancement/promotion coupled with a training program for entry level software specialists can result in a high retention rate for employees.

Using Computer Simulations to Enhance Problem Solving Skills in Science

by
Edward Vockell and Robert Rivers
Purdue University Calumet
Hammond, IN 46323

Science educators have suggested that students learn scientific principles and scientific processes most effectively by becoming actively involved in science. This suggests the active manipulation of objects while students carry out experiments to test hypotheses which they themselves generate. However, in many cases such active involvement of the students in experimental procedures is impossible because such experiments would be dangerous, time consuming, excessively expensive, or too technical or complicated.

The microcomputer provides an opportunity to enable students to perform many of these otherwise difficult experiments. When they use appropriately designed computer simulations, students are able to state a hypothesis, collect data to verify or reject the hypothesis, and replicate experimental procedures as often as necessary to come to a complete understanding of the scientific principles involved.

Although many computer simulations exist in areas related to the biological sciences, none were found which integrate these simulations into the existing biology curriculum. The present study employs a series of five simulations which *were* integrated with pretests, teacher's manuals, and student workbooks in such a way as to make the computer simulations an integral part of the curriculum.

In addition, an important goal of science programs is to help students develop more generalized scientific problem solving skills. Such skills would logically be developed more effectively if students were able to participate more intensively in a larger number of controlled experimental settings. It seems possible that the use of computer simulations would provide an opportunity for such increased problem solving practice.

Finally, it seems that a *guided* discovery approach would maximize the effectiveness of the computer simulations by focusing student attention on the most useful strategies at appropriate times. The present study was undertaken to determine:

1. Whether five separate computer simulations could accomplish educational goals as effectively as traditional modes of instruction.
2. Whether the use of such computer simulations provided a generalized increase in problem solving capabilities.

Methods

The computer simulations were the Computer Biology Labs developed by Luncsford, Rivers, and Vockell (1982). Specific programs were PLANT, OSMOSIS, MONOCROSS, MOTH, and BALANCE. PLANT enables students to perform experiments involving the growth of a plant under different amounts of light. OSMOSIS simulates the expansion and contraction of a red blood cell in different saline solutions.

MONOCROSS simulates the genetic reproduction of monohybrid genetic crosses. MOTH simulates the reproductive patterns of the pepper moth, typified in post-industrial England. BALANCE simulates the preditor-prey-environment relationship in nature, using deer as prey and the wolf as predator.

Each program came in a Guided or Unguided version. The difference was that the Guided version included approximately 100 words of guidance describing a useful strategy for applying the scientific method to that simulation. Three of the six experimental classes received the Guided version at random, and the other three received the Unguided version.

Performance with regard to each of the individual unit objectives was measured with the testing material which accompanied each unit. These tests were based on the specific behavioral objectives which were provided to both the experimental and control students. In addition, all the students took as a pretest and as a posttest the Test of Integrated Process Skills (TIPS) (Dillashaw & Okey, 1980) and the Watson/Glaser Test of Critical Thinking. The Watson/Glaser form ZM was used as a pretest and form A as a posttest.

The school in which the study was conducted was a large, predominantly black school in an industrial metropolitan area. The school had four biology teachers and 12 biology classes. All four teachers agreed to participate in the experiment. However, during the year in which the study was conducted, for reasons of health entirely unrelated to the experiment, two of the teachers were forced to drop out of the program and their replacements were unable to participate in the program. This resulted in complete pretest-posttest data on six classes (3 experimental and 3 control) and only partial data on the other six classes.

At the beginning of the school year each of the four teachers received two hours of training on how to use the computer simulations. The 12 biology classes were then assigned at random to either the experimental or the control conditions. The pretest of the TIPS and Watson/Glaser test were administered during the first two weeks of school, and those who were still available for the posttests took them during the last two weeks of school. In addition, within one to three days before each of the individual units, the students were administered the pretest on that unit; and then one to three days after each unit the students were administered the same test as a posttest on that unit.

Results

The results of the pretest-posttest analyses for each separate simulation indicated a pattern of significant differences among groups. In each case, the students using the guided version of the computer simulations scored higher (gained more) than either the control group or the students who used the simulations without guidance. In most (but not all) cases, the control group scored higher than the unguided computer simulations group. The superiority of the guided computer students was statistically significant in three out of five cases. When cumulative gains were taken into consideration, however, the superiority of the guided experimental group was statistically significant in all five cases. (This cumulative effect caused the guided students to gain more rapidly than other students on each subsequent posttest. This itself was an important outcome of using these simulations.)

In addition, the guided experimental students showed significant gains over the other two groups on both the TIPS and the Watson/Glaser test. These results indicate that the students who used the computer simulations which employed a guided discovery approach both mastered the designated course objectives as well as (or better than) the other students and developed problem solving abilities which generalized beyond the immediate objectives of the course.

Educational Importance of the Study

The following conclusions emerge from the present study:

1. Computerized simulations available for microcomputers can successfully accomplish the goals achieved within the normal biology curriculum. (Note that the "normal" curriculum itself was highly effective. That program took students from the 20th to the 60th percentile on the Watson/Glaser test.)
2. There is substantial evidence that the computer is able to accomplish many of these goals more effectively than the traditional approaches.
3. Students participating in such educational simulations were able to achieve a degree of problem solving capability which did not normally occur within the traditional curriculum. This generalized problem solving capability seems to have arisen from increased opportunities to practice such strategies in realistic settings.
4. The results strongly suggest that the computer is an important tool in teaching scientific problem solving skills only if suitable programs are accompanied by appropriate guidance.

References

Dillashaw, F.G., and Okey, J.R. Test of Integrated Science Skills for secondary science students. *Science Education*, 1980, *64*, 601-608.

Luncsford, D., Rivers, R., and Vockell, E.L. *BALANCE: The Predator/Prey relationship in Nature*. West Lafayette, Indiana, Diversified Educational Enterprises, 1982.

Luncsford, D., Rivers, R., and Vockell, E.L. *MOTH: The Evolution of the Pepper Moth*. West Lafayette, Indiana, Diversified Educational Enterprises, 1983.

Luncsford, D., Rivers, R., and Vockell, E.L. *MONOCROSS: A Simulation of Mono-Hybrid Genetic Crosses*. West Lafayette, Indiana, Diversified Educational Enterprises, 1983.

Luncsford, D., Rivers, R., and Vockell, E.L. *OSMOSIS: Osmotic Conditions in Red Blood Cells.* West Lafayette, Indiana. Diversified Educational Enterprises, 1983.

Luncsford, D., Rivers, R., and Vockell, E.L. *PLANT: A Plant Growth Simulation.* West Lafayette, Indiana. Diversified Educational Enterprises, 1983.

Watson, G., and Glaser, E.M. *Watson-Glaser Critical Thinking Appraisal.* New York: Harcourt Brace Jovanovich, 1964; Revised, 1980.

A GENERIC INSTRUCTIONAL MANAGEMENT SYSTEM DEVELOPED BY EDUCATORS FOR EDUCATORS

Lee J. Rodel
Evaluation Center
Hopkins Public Schools
Hopkins, MN 55343
Tel: (612)-933-9230

ABSTRACT

The Mastery Management System, a proven, generic, microcomputer based instructional management system developed by educators at the Hopkins Evaluation Center is described. Mastery Management monitors and reports both individual student and student group progress on identified instructional competencies, skills, or objectives. Developed for the Apple II microcomputer, the system provides a framework into which you enter your objective, test, and student information. Student test responses are scored and objective mastery recorded. An optional study helps feature may be used to prescribe appropriate study activities to students based on non-mastered or partially mastered objectives. Mastery Management has been implemented in over forty sites across the country in applications ranging from elementary school math and reading basic skills to secondary school science and social studies.

INTRODUCTION

The increasing demands for education accountability are being heard across all subject matters and grade levels in our nation's schools. Such demands are coming from parents, our taxpaying community, federal and state governmental bodies, and from students themselves.

A criterion referenced testing model consisting of specifying learning objectives or skills, teaching to these objectives and then testing the student achievement of these objectives is being used by educators to respond to the demands for accountability. To use such an achievement monitoring model, however, requires considerable effort and time on the part of the teacher. Even after learning objectives are written, test items related to those objectives developed, and paper and pencil tests designed and developed testing a subset of the objectives, such tests must be administered to students, scored, and individual student and student group performance data related to objectives produced and stored. This test scoring and recordkeeping task in a criterion-referenced model can only be efficiently and effectively done with the aid of a computer. There is need for a recordkeeping tool to assist the educators in the management of instruction -- determining and reporting student mastery of objectives and designing instruction to facilitate further student mastery of these objectives.

The Hopkins, Minnesota, Public Schools have been involved for over ten years in the development and use of computerized tools to aid the management of instruction. The Hopkins Public Schools, a suburban Minneapolis K-12 school district of approximately 7000 students, have developed learning objectives in grades 2-12 for most subject matters including math, science, social studies, and language arts. Tests covering these objectives have been developed and administered on a regular basis. Computerized recordkeeping support to score tests and report results has been developed on both large main frame computer and microcomputer systems. CAM, Comprehensive Achievement Monitoring, operating on the Apple II, was the first microcomputer instructional management support tool designed, developed, and distributed by the Hopkins Schools Evaluation Center. Over 90 sites across the country adopted the CAM system on the Apple II to assist monitoring of established performance criteria.

MASTERY MANAGEMENT SYSTEM

Based on our own use of CAM on the Apple II and the feedback received from the many CAM users outside the school district, the Evaluation Center designed and developed another instructional management tool also operating on the Apple II microcomputer. Called the Mastery Management System, this tool, developed by educators for educators, is a generic microcomputer based instructional management system which monitors both individual student and student group mastery of user defined objectives based on user defined mastery criteria. The Mastery Management System is a framework into which the user enters his objectives, test answer keys, and student information. Student test responses are then scored and objective mastery recorded and reported as desired. An optional study helps feature relates to objectives the instructional resources/activities available to the user in his school or classroom. The study helps feature can be used to prescribe appropriate study activities to students for on non-mastered or partially mastered objectives.

To date the Mastery Management System has been implemented in approximately 40 sites across the country in classroom, school and district wide applications to monitor individual student, class, school, and district-wide mastery of individual and/or groups of objectives.

The remainder of this paper provides additional detail on the capabilities and use of the Mastery Management System.

SYSTEM DEFINITION

A. Parameters (per data diskette)

- Objectives - 400 maximum with mastery levels and optional study helps
- Test Answer Keys - 50 maximum, unlimited with replacement
- Questions - 50 questions per test answer key
- Teachers/Student Groups - 20 maximum
- Students 900 maximum

B. Hardware/Software

- Apple II, II Plus IIe with minimum 48K user memory
- Applesoft BASIC in ROM
- Dual Disk Drives, 3.3 DOS
- Hard copy printer (132 characters width capability)
- Optional card/page reader for student response entry

C. Diskettes

- MMS Program Master Diskette
- Course Data Diskette
- Optional Study Helps Diskette

The Mastery Management System is menu driven and is highly user friendly. By responding to prompts displayed, the user of the Mastery Management System selects the processing operations options to enter objectives, answer keys, students and study helps (optional). Once this information is entered, student test responses may be entered by keyboard or card/page reader and scored.

REPORTS

Reports available are:

A. As part of test scoring run:

- Student Coupon (with or without study helps) - scores student responses by question and shows student mastery status of objectives tested.

- Test Form Evaluation - shows group performance question by question on test.
- Teacher Gradebook Report - shows total test score and objectives mastery data for each student in class.

B. As stand alone reports upon request:

- Student Status Report - contains selected mastery status information for selected individual students on selected objectives.
- Study Helps Reports - contain study helps for selected individual students for not mastered and partially mastered objectives.
- Objective Mastery Reports - summarize for the teacher the selected mastery status of selected students on selected objectives.
- Student Progress Reports - contain for selected students on selected objectives the current percent mastered, partially mastered, not mastered, and not yet assessed.
- Summary Reports - summarize the mastery status for selected student groups on selected objectives.

The sample Student Coupon below shows information provided to the student after each test:

```
*********************************************************************
*  AGRIMBEN BRUCE 100     COURSE: MATH3    FORM: 1         10/25/82 *
*  316314             SCORE: 37/50( 74%)        TEACHER: D. SCHULTE *
*===================================================================*
*  Q#  OBJ    ! Q#  OBJ    ! Q#  OBJ    ! Q#  OBJ    ! Q#  OBJ      *
*  1   102  - ! 11  107  + ! 21  114  + ! 31  206  + ! 41  304  +   *
*  2   102  - ! 12  107  + ! 22  114  + ! 32  206  + ! 42  304  +   *
*  3   104  + ! 13  108  + ! 23  202  - ! 33  208  + ! 43  306  +   *
*  4   104  - ! 14  108  + ! 24  202    ! 34  208  + ! 44  306  -   *
*  5   104  + ! 15  110  + ! 25  202  - ! 35  208  + ! 45  306  -   *
*  6   105  + ! 16  110  + ! 26  203  + ! 36  210  + ! 46  402  +   *
*  7   105  - ! 17  110  + ! 27  203  + ! 37  210  + ! 47  402  +   *
*  8   106  + ! 18  110  + ! 28  204  + ! 38  212  + ! 48  404  -   *
*  9   106  + ! 19  112  + ! 29  204  + ! 39  212  + ! 49  404  -   *
*  10  106  - ! 20  112  + ! 30  206  + ! 40  212  + ! 50  404      *
*                                                                   *
*========================MASTERED OBJECTIVES========================*
*     107 ZERO CONCEPT +   108 USE OF OR-       110 USE<>=IN + OR = *
*     112 KNOWS +- FACTS   114 2STEP EQUATIONS  203 CHECK ADDITION  *
*     204 2ADDENDS 3DIGIT  206 SUBTRACT 2DIGIT  208 SUBTRACT 3DIGIT *
*     210 CHECK SUBTRACT   212 WRITE PROB VERT  304 REGROUP 2ADDEND *
*     402 -REGROUP 2DIGIT                                           *
*                                                                   *
*-------------------PARTIALLY MASTERED OBJECTIVES-------------------*
*     104 LABEL PROB PART  105 SUBT/VOCAB       106 COM PROP/ADD    *
*                                                                   *
*----------------------NOT MASTERED OBJECTIVES----------------------*
*     102 +- HORIZ/VERTIC  202 3ADDENDS 2DIGIT  306 REGROUP 3ADDEND *
*     404 -CHECK/REGROUP                                            *
*********************************************************************
```

The following sample Teacher Objective Report shows for the teacher the students who have not mastered (N) or only partially mastered (P) a subset of course objectives.

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< TEACHER OBJECTIVE REPORT 01/06/83 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    TEACHER: D. SCHULTE      MATH3 / OBJECTIVES 102 TO 1118 (40 OBJS.) PARTLY MASTERED, NOT MASTERED INFO
OBJECTIVE:    0 102 +- HORIZ/VERTIC 0 114 2STEP EQUATIONS 0 304 REGROUP 2ADDEND 0 702 1STEP WORD PROB 01004 COM/ASSOC/MULT
              0 104 LABEL PROB PART 0 202 3ADDENDS 2DIGIT 0 306 REGROUP 3ADDEND 0 704 SOLVE WORD PROB 01006 VERBALIZE +/X
              0 105 SUBT/VOCAB      0 203 CHECK ADDITION  0 402 -REGROUP 2DIGIT 0 805 REWRITE +- PROB 01008 RELATE MULT/DIV
              0 106 COM PROP/ADD    0 204 2ADDENDS 3DIGIT 0 404 -CHECK/REGROUP  0 808 MISSING ADDEND  01110 X0/0 PRODUCT
              0 107 ZERO CONCEPT +  0 206 SUBTRACT 2DIGIT 0 502 GROUP 3 ADDENDS 0 810 ADD SUBT MONEY  01112 USE DIV SYMBOL
              0 108 USE OF OR-      0 208 SUBTRACT 3DIGIT 0 504 +REGROUP 3DIGIT 0 811 SUB MONEY BORRO 01114 DIVIDING BY ONE
              0 110 USE()=IN + OR = 0 210 CHECK SUBTRACT  0 602 -REGROUP 3DIGIT 0 812 REGROUP MONEY   01116 DIV 0 BY ITSELF
              0 112 HNGMS +- FACTS  0 212 WRITE PROB VERT 0 604 -3DIGIT/ZEROS   01002 RELATE ADD/MULT 01118 /FACTS TO 45-9
PAGE 1
                                                                                   1 1 1 1 1 1 1 1 1
                          1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 4 4 5 5 6 6 7 7 8 8 8 8 8 0 0 0 0 1 1 1 1 1
                          0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1
ID#  STUDENT      SCH. ID#  2 4 5 6 7 8 0 2 4 2 3 4 6 8 0 2 4 6 2 4 2 4 2 4 2 4 5 8 0 1 2 2 4 6 8 0 2 4 6 8
_______________________________________________________________________________________________________________
 100 AGRIMSEN BRUCE     316314 N P P P . . . . . N . . . . . . . N . N . . . . . . . . . . . . . . . . . . . .
 101 AUANT JAMES        843542 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 102 BECK STEVEN        967237 P N N N N N . P N N N P N N N N N N P P . . . . . . . . . . . . . . . . . . . .
 104 CARR TERRY         220937 N P . P P . . N P N N P P N P N N P P N . . . . . . . . . . . . . . . . . . . .
 106 CORNETT DOYLE      675841 N P . P P . . N P P . N N P P P P N P N . . . . . . . . . . . . . . . . . . . .
 108 GALLAGHER JO       404344 P P . N . P N P P N . P P P P P . P . P . . . . . . . . . . . . . . . . . . . .
 110 GRIMES BOB         270873 P N N N N P N N P N N P N N . N N N N P . . . . . . . . . . . . . . . . . . . .
 112 HELMEN JOHN        835743 . . P . P N . N . P N . P . P P . P . . . . . . . . . . . . . . . . . . . . . .
 114 HIXON JERRY        305791 P P . P N P P P . P P P P P . P . N . N . . . . . . . . . . . . . . . . . . . .
 116 HARRISON DAN       451127 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 118 JACOBS DICK        563273 N . P N P . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 120 JOHNSON BRADLEY    653344 P N N N N . N N N N N N N N N N N N N N . . . . . . . . . . . . . . . . . . . .
 122 KARNOW TERRY       782544 P P N N P . N N P P P P N N N N N N P N . . . . . . . . . . . . . . . . . . . .
 124 KRUEGER DONNA      495614 N P P N P P P P . N N P P P P N N N P P . . . . . . . . . . . . . . . . . . . .
 126 LEE HEATHER        134163 . P . P . P . N . P P P P . . P . P . . . . . . . . . . . . . . . . . . . . . .
 128 LEWIS JERRY        384333 P P . P P P . N P P . P P P . P . P . P . . . . . . . . . . . . . . . . . . . .
 130 MCQUILLEN DANIEL   791841 N . P . P . P P P . P P P . P P . P . P . . . . . . . . . . . . . . . . . . . .
 132 MASON JAMES        113452 . . . P . P P P P P . P . N . N . P . P . . . . . . . . . . . . . . . . . . . .
 134 MC GUIRE LEE       489344 P P P . P N P N P P N . P P . P P . P . . . . . . . . . . . . . . . . . . . . .
 136 OSTENSON JACK      677814 . . . . . . . . . N N P N N N N N N N N . . . . . . . . . . . . . . . . . . . .
 138 OVERBY MARK        962214 . N P P P . . . . . N . P . . P P . . . . . . . . . . . . . . . . . . . . . . .
 140 PETERSON KEN       131844 P N . N . P P P P P N P N P . N . . N . . . . . . . . . . . . . . . . . . . . .
 142 PORTER MARIE       147228 . P . P . P . P P . P . P N . P . P . P . . . . . . . . . . . . . . . . . . . .
 144 REDMAN GEORGE      590737 P P P . P P P P . . . P P . . N . P P P . . . . . . . . . . . . . . . . . . . .
 146 RICHARDSON SUE     811744 P N . N . P N N . P P N N N N N N N P N . . . . . . . . . . . . . . . . . . . .
 148 STEPHENS RUSSELL   394273 P P N N P P N P P N N . N N N N N N N N . . . . . . . . . . . . . . . . . . . .
```

The Evaluation Center is prepared to provide training on the use of the Mastery Management System and support users in their implementation of instructional management systems in support of criterion-referenced student achievement monitoring and reporting systems.

Further information and awareness materials on the Mastery Management System may be obtained by contacting:

Evaluation Center
Hopkins Public Schools
1001 Highway #7
Hopkins, MN 55343
Tel: (612) 933-9230

PERSONAL COMPUTER PLAN
AT
STEVENS INSTITUTE OF TECHNOLOGY

Edward A. Friedman
M. Peter Jurkat
Joseph J. Moeller, Jr.
Roger S. Pinkham
Bernard Rosen
Stevens Institute of Technology
Castle Point Station
Hoboken, New Jersey 07030

ABSTRACT

Stevens Institute has required all freshmen entering the Science curriculum and the Systems Planning and Management curriculum in Fall 1982 to own a microcomputer system. Stevens is the first major college in the country to introduce such a requirement. Through a variety of curriculum-based activities, students will use the computer as an integral part of their intellectual support system. In addition, these students will continue to have access to the major computing resources on campus. The Personal Computer Plan is designed to increase the "computer fluency" of students through a continuous integration of personal computer use throughout the undergraduate curricula.

Within the next few years the 3 R's in the primary and secondary schools will become computer literacy plus 3 R's. For the last few years, universities, and in particular the scientific and engineering colleges, have realized that they must provide opportunities for students to develop computer literacy. In addition, there must be a focus on what we, at Stevens, call computer fluency. By the latter term we mean the knowledge of the collection of programming techniques, algorithms, modelling methods and so on that one must know to make effective application of computers in science and engineering.

At Stevens we were caught between the desire to increase our students' use of the computer for their professional training and the finite computer resources available to us. That we wished the students' experience with the computer to be an interactive one made the situation even more difficult. During considerations of this problem the suggestion was made that each student should be required to own a personal computer. While the idea seemed a bit far-fetched when it was first proposed in 1980, the rapid advances in personal computer hardware (and its decreasing cost), the growing acceptance of personal computers by the public, and greater faculty familiarity with the power of these machines made it possible for us to adopt this plan for all incoming freshman in the Science and the Systems Planning and Management (SP&M) curricula in the Fall of 1982. Among the advantages we foresaw in this plan were:

(1) The easy accessibility to the computers would encourage the students to use them as naturally as they then used their calculators.

(2) The microcomputers were relatively easy to use as data acquisition devices and controllers. This particular use is of increasing importance in technology and hence should

be part of our students' training.

(3) The computing facility available to the incoming students could be upgraded with each freshman class.

Stevens Institute curricula, which include Engineering, Science, and SP&M, are relatively unspecialized. For example, our engineering students receive Bachelor of Engineering degrees upon graduation that do not specify a particular engineering discipline. The core courses in each of these curricula constitute a large part of the program that each student must take. This core made possible the introduction of the PCP into the Science and SP&M curricula in an orderly manner; the students could all take specific courses in which the personal computers were certain to be utilized. For example, in the Science curriculum every student takes the following courses:

MA11	Introduction to Programming	1st Term Freshman
MA12	Linear Algebra and Matrices	2nd Term Freshman
PH27	Computational Physics	2nd Term Sophomore

The decision to adopt this plan was not made until we chose a computer. Among the criteria influencing this choice were ease of use, graphics capability, display, keyboard, expandability of the system, I/O interface and cost. The last factor was important for the usual reasons but also because

(1) This was the first time such a requirement was being implemented.
(2) The time of announcement of the plan was after the students had decided to come to Stevens, and had paid their acceptance deposit.

The machine adopted was the Atari 800 with 16K RAM and a Basic cartridge. These items, together with a Sharp TV Receiver and an Atari 410 Digital Recorder provided the basic package each of the incoming students purchased at a cost of approximately $750 dollars. The acquisition of the microcomputer systems was made directly through the Institute.

Within this year some of the early thoughts on the Personal Computer Plan have already been modified. For example, we thought at first that portability would be important since we expected that the students would be required to bring the computers, sans TV set, to the laboratory. Now we expect to provide some computers specially modified for this purpose. What some of us feared would be a negative reaction from parents to the extra cost of the computer did not materialize; in fact some parents of engineering students expressed the wish that their children could also participate.

The students have now completed the Introduction to Programming course. Because each student possessed virtually unlimited access to computing facilities, that course could be taught with an emphasis on algorithms and programming style and with a great deemphasis on instruction in the elements of the Basic language. The individual student could reasonably be expected to master Basic from the manual just as well as the average purchaser of a home computer. As a consequence, these students were able to be instructed in such topics as the use of Monte Carlo methods to calculate PI and an integration technique for differential equations within the first seven weeks of the term.

At present (March 1983) we are in the process of selection of a computer system for next year, with the very strong possibility that all incoming freshmen--this means our engineering students also--will participate. Among the additional needs for a suitable system are the availability of FORTRAN (and hence disk drives), of software for engineering computer graphics, and communications and networking capability.

COMPUTER LITERACY

PLAN FOR MIDDLE MANAGEMENT AND ITS IMPACT ON DECISION SUPPORT SYSTEMS

Sara Rushinek, Ph.D.
Avi Rushinek, Ph.D.

University of Miami

Coral Gables, Florida 33124

ABSTRACT

This paper is an examination of some viewpoints on the subject of computer literacy. Its basic premise is that in comtemporary times a knowledge of the computer in terms of what it is, what it can do, and how it does it, is no longer a nicety. Computer literacy has become and will continue to be in increasing proportion, a necessary ingredient for a middle manager's success in a typical business enterprise. Exactly how important and indeed how very basic this knowledge has become is discussed and consideration is given to the ramifications of illiteracy as well as the barriers to achieving literacy.

COMPUTER LITERACY

To go back to the title of this paper, we contend that computer literacy is becoming as essential to the individual in a managerial role as his/her ability to read and write. Today he/she has at his/her disposal an extraordinary tool with the capability to process and convert enormous quantities of data into meaningful managerial information. Despite the power of this tool, to many managers it is as valuable as a literary masterpiece in the hands of one who cannot read.

A manager who is not computer literate is at a distinct disadvantage to the manager who is literate; and an industry or society that cannot take advantage of the tools available to it for lack of knowledge as their capabilities becomes non-competitive with those who can make effective use of their resources. Initially vulnerable, the computer illiterate manager will eventually be replaced by one who has the skills to make use of essential tools. This is not to suggest that the typical middle manager is in immediate peril of losing his/her livelihood. To the contrary, the extent of computer illiteracy is so prevalent that he/she is among the majority and is approaching the opportunity to become literate. Failure to do so however could imperil his/her comparitive affectiveness and upward mobility. But it should not be thought of as a critical situation whose remedy must be adminsitered to at once. In the same breath it must be recognized that educational institutions, in educating managers of the future, have begun to incorporate computer knowledge as part of a required curriculum. With the proliferation of micro-computers for home use, children at pre-teen years are beginning the familiarization process in an informal but effective way. This will inevitably lead to a greater need for more formal avenues of computer knowledge which will occur at earlier stages in the educational process.

The topic of computer literacy should not be viewed as a threat, but rather it should be viewed in the spirit of Victor Hugo's statemen, "There is nothing more powerful than an idea whose time has come".(Radar, 1982)

Computer literacy defined, is what one should know about the basic machine, and its related devices in order that it may best serve its user. In addition to knowing about the computer, one should know what to do with it. That is, for a greater appreciation of the computer, merely reading about the computer and what it does is insufficient. (Yarrish, 1982). There must be a "hands on" process where the manager has the opportunity to touch and work with the computer.

The most basic segment of literacy is What A Computer Is. This aspect deals with the central processing unit, its major functions and the term "hardware". This includes all devices and equipment necessary to perform the functions of data preparation input of data, computation, control of computation storage and output of results.

The second segment, What a Computer Can Do, deals with the concept of "software" which provides the computer with instructions that it can understand in order to carry out its tasks. This segment also deals with the computers' ability to process large amounts of data in short periods of time. To a manager, this aspect is of particular value for it provides him/her with information which otherwise would be both extremely time consuming and costly if this processing was done by people. This segment deals with the computers capabilities as well as its limitations.

The third segment is How the Computer Works and is the chapter title that led to the research for this paper. While this forms the final area in computer literacy, it probably is the least important of the three segments. Never the less it is one that manager should have a familiarity with although his/her intimacy would be greater with the first two aspects. The questions that arises is how much should a manager know about how a computer works. Mr. Edward Yarrish, managing associate of Executive Technology Associates, in an article published in April 1982 issue of the ABA Bank Journal answers the questions by an analogy. He states, "The vast majority of us have no need to understand vehicle technology as thoroughly as the engineer who designs vehicles or the mechanic who maintains them. But we should understand technology well enough to be intelligent users and purchasers". If follows therefore that if one is to purchase and use a computer as one purchases and uses an automobile, a fundamental knowledge of how the computer works is required. In that context, the subject matter in this area can be divided into three groups.

I. How the computer internally organizes and operates.

II. How the processing takes place in the instruction and execution cycles of the Central Processing Unit.

III. The Operating System and how this software monitors and controls all input/output and processing operations.

The concept of the bit and two state state binary representation are basic to "how the circuits switches and bits in the computer are linked to one another so that processing is possible an understanding of "fixed word Structure" "byte structures", "registers", and "accumulators" is necessary.

The second of the three groups identified, "how the processing takes place", concerns, itself with the steps the CPU takes in the instruction and execution phases. The "machine cycle", refers to the length of time the CPU requires to perform the two phases. In this area, the concept of "overlapped processing" made possible by "channels" and "buffers" is of importance. This is the understanding of how a computer speed is maintained as it compensates for input and output devices that function at slower speeds and makes possible simultaneous input, processing and output operations.

The third group, "the operating system", is a set of programs that the manufacturer of a CPU provides and it is this software tool that monitors and controls all input/output and processing operations. A very important aspect is the "service functions" the operating system performs. In this area references is made to the "translator programs". Their purpose is to convert programming languages into machine language or electronic pulses that the computer system understand (Senn, 1981). Having defined the subject matter of computer literacy there are barriers that exist which have hindered the attainment of a greater level of computer literacy among managers.

In addition to the fear of computers due to misconception there is a fear according to Joseph Judenberg, Manager of Management Information Systems for New Yorks Chase Manhattan Bank that when people sit down at the computer terminal, "They worry that they will look stupid,- that they can't master the techniques". There is also a further concern or fear that is experienced having once accomplished the task of overcoming fears. This is known as the fear of becoming a computer junkie or the hammer principle". This principle states that if you give someone a hammer he/she will use it. People can become so involved with the computer terminal that it takes over their day.

Nonetheless, fear as a barrier to computer literacy must be dealt with. In addition to formal training, which is becoming more widely available in the classroom, industry has begun to introduce internal programs to foster computer literacy within their companies. One way of accomplishing computer literacy is to develop a corporate strategy. The following is an example of five point strategy to create computer literate managers (Yarrishe, 1982).

1. Increase the learning opportunities for people to understand computer technologies. This should be done through simultaneous, multiple avenues, including hands-on training, availability of current technological publications, and printed poster in highly visible locations.

2. Increase the rate and amount of direct hands-on computer exposure. Look to small, inexpensive personal computers as a way to accomplish this.

3. Make the technology easier to acquire. The same flexibility and variety will apply to computer equipment as now applies to office furniture and typewriters.

4. Encourage computerized experimentation. When people do not have experience with computers, they will make mistakes while learning. Expect mistakes and use then to your advantage. Ask: "What did we learn from that?" and "How can we do it better next time".

5. Continously adapt the new technologies to the realities of human behavior. Bend the machines to fit the people. It never works the other way.

At the outset of this paper we expressed the notion that the manager at the middle level is in most need of literacy. In making this statement, we are then speaking of the largest segment of management in terms of sheer number. But it is this group who must react on a daily basis on an operating level. As such, it is this level of management that receives the bulk of the computers output and is responsible for taking actions and making decisions that this information will indicate. A manager who is not computer literate will be able to understand how the information was developed and to the extent that it must be modified, enlarged, summarized, and supported, he/she will be equipped to do so. Futhermore it is the middle management of a firm that aspires to top management positions. Therefore, as computer literacy is gained on the middle management level today, it is carried forward to the top management tomorrow. Indeed it is a requirement at all levels but the greatest practical day to day benefit belongs to those who function at the middle management level.

With this we must introduce a caveat. The computer as tool can and does provide the manager with information. However, this information has value if it is used properly and the question as to how effectively the information enters into the decision arises.

Rodney H. Brady, Vice-President of Aircraft Division, Huges Tool Company, states that according to a careful study the impact of the computer in the decision making process has been limited. "If the computer has had any impact on the top management decision making process, the effect has not been directly on top management, but on the manner in which middle managers are contributing to top management decision making." The thrust of Mr. Brady's article deals with top management's involvement with the computer and decision making. We however, question whether middle management, even if computer literate has a knowledge of how to properly evaluate computer output and incorporate the interpretation of this output in a systematic way when making decisions. Perhaps this area goes beyond the subject of computer literacy per se. On the otherhand, we feel that while being computer literate is necessary for a manager, he/she must go beyond that. Once having mastered the basics that forms computer literacy, he/she must then know how to properly make use of the output with which he/she, through computer literacy has become both more conversant and intimate.

SUMMARY CONCLUSION AND IMPLICATIONS

This paper reports on the subject of computer literacy. It discusses the meaning of the term, its components, and its importance to the middle manager. In particular, the barriers one faces in promulgating computer literacy, and its impact on decision making.

We conclude that the typical middle manager will be unable to compete unless he/she becomes computer literate and that computer literacy is no longer a nicety. It has become a necessity and is not confined to a manager in a business enterprise. It is necessary in all walks of life and should be introduced in our educational system at early levels.

REFERENCES

1. Anonymous, "End User Curricula Development-Practical Application" (Cargill Incorporated, 1982)
2. Brady, Rodney H. "Computers in top-level decision making" Computer Management Series, (reprinted from HBR July-August 1967 No. 67402)
3. Lawson, Harold W., Jr, "An Approach to Improving Computer Literacy" Microprocessing & Microprogramming, (Mar 1982, pgs 123-132)
4. Woodruff, Jary, "DP Illiteracy Held Future Handicap" Computerworld, (June 4, 1979)
5. Rader, Louis T. "Computer Literacy, Option or Imperative" Vital Speeches of the Day, (June 1, 1982)
6. Senn, James A., "Information Systems in Management" (Wadsworth Publishing Company, 1982) pgs. 133-148
7. Vazsonyi, Andrew "Information Systems in Management Science" Interfaces (Feb 1982) pgs 74-78.
8. Yarrish, Edward B. "Computer Literacy: a new requirement for managers?" ABA Banking Journal (April 1982)

COMPUTERS IN EDUCATION: BOOTING UP A UNIVERSITY

Pat Semmes

Jean Wilson

Trinity University
715 Stadium Dr.
San Antonio, Texas 78284

At Trinity University faculty in the Education Department are working with faculty in the Computing and Information Sciences Department to integrate computer education into existing programs for preservice and in-service teachers.

Trinity University is a small, liberal arts school with high academic standards located in South Texas. Academic departments within the University include a Department of Education which is part of the Division of Behavioral Sciences and a Computing and Information Sciences Department which is in the Division of Business. The University has undergone dramatic change in interest and action in integrating computer education into the existing programs for preservice and in-service teachers. In addition, new ways in which the University can work with area school districts to help them in implementing computers in the classroom are likely to be put into effect at the end of the current semester. Recognizing the need in the schools for information related to appropriate uses of computers in education, faculty in Education and faculty in Computing Science have made a joint commitment to fill this need. Special computer courses taught in the Computing Science Department for those interested in education have been offered for the first time this year and are being taught by a faculty member with a strong interest in computer education and whose background includes both education and computing science (Semmes). The integration of computer experiences into the Education curriculum on a broad scale requires considerable faculty development; the Education faculty member whose field is mathematics education (Wilson) has taken the primary responsibility for coordinating computer education in her department.

Having a single course which would cover all of the computer uses which relate to each of the education courses would have required expertise on the part of only one faculty member in Education. However, the faculty believed that it would be more valuable for the students if each faculty member could show how the computer could be used in his/her subject area. Moreover, faculty members believed that learning about computers would be an important part of their own development. According to Bell (1982), "An effective way to get uninitiated school and college teachers on the microcomputers is to pay them to attend training seminars, without requiring homework....Experience indicates that for 10 trainees,...2 will do instructional computing in their classes." Although there has been no compensation for faculty development in computer education, about half of the Education faculty have taken at least one course in computing and several are currently taking their second course. Three faculty members have already had papers published (excluding this one) related to computers in education. In short, the faculty has made a strong commitment to providing computer education for teachers. However, faculty commitment is not sufficient to insure a successful program. Such a program would seem to require adequate computer software and hardware.

Although the Department of Education does not have a computer facility, students in courses in Computing and Information Sciences (C&IS) have access to terminals to an IBM 3031 and to a microcomputer laboratory which houses six Apple II's. It is likely that a microcomputer laboratory with 18 IBM Personal Computer's and six Apple II's will be in place soon and will be available to the University as a whole. The lab will be located near the Computing Science Department which is on the other side of the campus from Education and will service computing science students as well as others. However, to meet the special needs of Education, it would be desirable to have a sample cluster of computers as well as a set of one particular brand available in the Education Resource Center. Since Education will have access to the new microcomputer lab, a sample cluster has become the highest priority.

In the Fall '82, the Dean of Behavioral Sciences placed high on his list of priorities for capital equipment a request from the Education Department for the purchase of an Apple II, a Radio Shack, and a Commodore PET. At that time, the President received so many requests for microcomputers that he declared a moritorium in order to study the situation. What was especially disconcerting to the president was the variety of microcomputers requested and the lack of a centralized location, typical concerns for an administrator. A University Committee was formed and faculty members were asked to respond to the suggestion of a centralized microcomputer laboratory having 24 IBM PC's.

Semmes in Computing Science argued that preservice and in-service teachers and faculty, as well, need experience with a variety of microcomputers. She pointed out that the main microcomputers currently in use in the schools at the elementary and secondary levels are the Apple, Radio Shack, Commodore, Atari, and Texas Instruments and noted, "It is not important for perspective teachers and in-service teachers to be intimately knowledgeable about each microcomputer on the market, but it is important that the Trinity faculty have a variety of machines to demonstrate so that 'students' can experience equipment similarities and differences. Moreover, when working with inservice teachers, faculty need to be able to relate to needs for help with whichever brand computer the schools already have. Neither student nor faculty needs for hands-on experience with the various machines can be adequately met by working with only one brand machine even if that is coupled with trips to computer stores and/or reading in the library."

Wilson in Education responded that "In order for the Department of Education at Trinity University to take a leadership role in this community and region, we need to provide experiences for our students at the undergraduate and graduate levels to become computer literate and to provide in-service workshops to enable teachers to acquire the functional knowledge of computers and how to integrate computers into the curriculum." She enclosed the results of a study of the brands and uses of microcomputers in twelve school districts in the San Antonio area. She noted that there are about 5,000 microcomputers currently (as of Fall '82) in place in the San Antonio area and stressed the need for a 'sample cluster' of the computers that are in the schools today. These comments as well as remarks by others probably resulted in the decision to include six Apple II's in the proposed microcomputer lab. Bids for equipment have been solicited; however, a final decision about the lab has not been made.

Currently, it appears unlikely that Education will be provided with any computers for the Education Resource Center through capital equipment funds. However, because of the dire need for equipment that is being created by commitments to provide computer education for teachers as part of several up-coming programs, equipment will be obtained!

Recently, the San Antonio ISD's board agreed to a joint program with Trinity in which the University faculty will work with teachers and administrators in a pilot study of ten elementary schools in which computers will be implemented at the third, fourth, and fifth grade levels. The project will begin this summer and will be in three phases: the first in which principals and administrators will receive training, the second one in which teachers will be trained, and the third in which teachers will implement, under University faculty supervision, computer use in their classrooms. Another project planned for the summer is a four day computer education workshop for teachers in school districts which participate in the Trinity University Gifted and Talented Consortium. In March a symposium on computer education for teachers of the gifted and talented was held. Teachers in this program attended a one day workshop on using computers with gifted and talented children. Semmes and Wilson use their own microcomputers when necessary, and some teacher training will be done using a school's own equipment. Although the lack of equipment is disappointing, the feeling of excitement concerning teacher education related to computers is growing.

In summary, the University's commitment to providing teacher training in computer education has changed drastically during the past year. Two courses in the Computing Science Department are now being offered which are appropriate for students and faculty in Education. Approval is being sought for a program of work to enable students to become certified to teach computing science at the secondary level. The Education faculty has made a commitment to become educated in the use of computers and to integrate the computer into course offerings. Several important projects involving in-service teacher training have been initiated by the Education Department. Sources for funding a microcomputer center in Education are actively being sought. It would appear that the faculty's goal of meeting the computer education needs of preservice and in-service teachers is beginning to be realized.

REFERENCES

Bell, Frederick H., "Implementing instructional computing and computer literacy in a school or college." AEDS Journal, September 1982.

A DECISION SUPPORT SYSTEM FOR DETERMINING EQUALITY OF EDUCATIONAL OPPORTUNITY

Porter Sexton, Ph.D.
Management Information Services
Portland Public Schools
P.O. Box 3107
Portland, Oregon 97208

ABSTRACT

How can school districts ensure that they are providing an equal opportunity for educational achievement to all their students? A group of key administrators in the Portland Public Schools was convened to address this question. The result was a plan of research which led to the author's Ph.D. Dissertation[1] on the development of a computer system which calculates an EEO Accountability Index for a school system under various scenarios. This paper describes the development and potential use of such a Decision Support System. The audience for this paper might be (1) information specialists responsible for the design of Decision Support Systems, (2) school district board members, superintendents, and decision makers, and (3) other academicians interested in developing a better understanding of the concepts of EEO and the implications of those concepts for education decision making.

INTRODUCTION

The Federal Elementary and Secondary Education Acts of 1965 and 1972 established definitions, guidelines, statutory requirements, and funding aimed at Equality of Educational Opportunity (EEO). A myriad of programs to improve, monitor, and evaluate educational opportunity has been developed since that time. Associated with this development has been an increasing burden on school districts to effectively meet the attendant information requirements, such as program monitoring, evaluating, and reporting.

Clemson[2] suggests that a computerized Decision Support System (DSS) could be designed and implemented to assist decision makers with questions specifically related to equality of educational opportunity. If so, such a system could provide a great deal of support to the school district in meeting its EEO information needs.

There are two aspects of the task of designing such a DSS which make it particularly difficult, and which changed it from one simply of design to one of research. Those aspects are a) there is no theoretical agreement about the basic concepts of EEO[3], and b) therefore the decision makers and ultimate users of the DSS may not be able to identify key components or relationships from which to construct a basic model.[4] Thus, the design of such a DSS requires not only the construction of a new model, but, in addition, it requires the development of a methodology for the construction of such a model.

SYSTEM DESIGN

The author undertook the task of designing such a DSS with a system design process which consisted of the following five major phases:

Phase 1: The Task Group

The support of the local school district was obtained, and, with the assistance of the superintendent, a task group of concerned administrators was assembled to assist in defining the needs and goals for the DSS.

Phase 2: Key Variables

Working with the task group, the author developed a list of key variables to be included in the support system and gathered and stored the data for a preliminary version of the DSS. Variables examined included cost-per-student, pupil-teacher ratios, school minority percentages, school faculty minority percentages, reading test scores, grades, attendance, suspensions, expulsions, indicators, socio-economic status, support of local community, school climate, teacher expectations, and attitude indicators.

Phase 3: Models of EEO

A thorough review of the literature was undertaken in order to reach an understanding of the concepts of Equality of Educational Opportunity (EEO) sufficient to provide a model to be included in the DSS. The literature review led to the identification of four conceptual (subjective) models of EEO. The four models tended to emphasize, respectively, (1) tangible aspects (e.g., cost-per-student), (2) intangible aspects (e.g., teacher expectations), (3) effectiveness aspects (e.g., test score gains) and (4) achievement aspects (e.g., test scores for competency).

Phase 4: Analyses

The four subjective models of EEO identified from the literature were proposed as hypothetical models for inclusion in the DSS. Analytical techniques were then undertaken to determine which model, if any, best represented reality as determined by the data gathered in Phase 2. These analytical techniques included correlation analysis, subgroup analysis, graphic analysis, factor analysis, and phenomena explanation. The results of these analyses helped to identify the aspects of the subjective models which best fitted the data. The analyses led to the development of the EEO Accountability Model (shown below), so named because of the system accountability brought about by various "feedback loops" in the model.

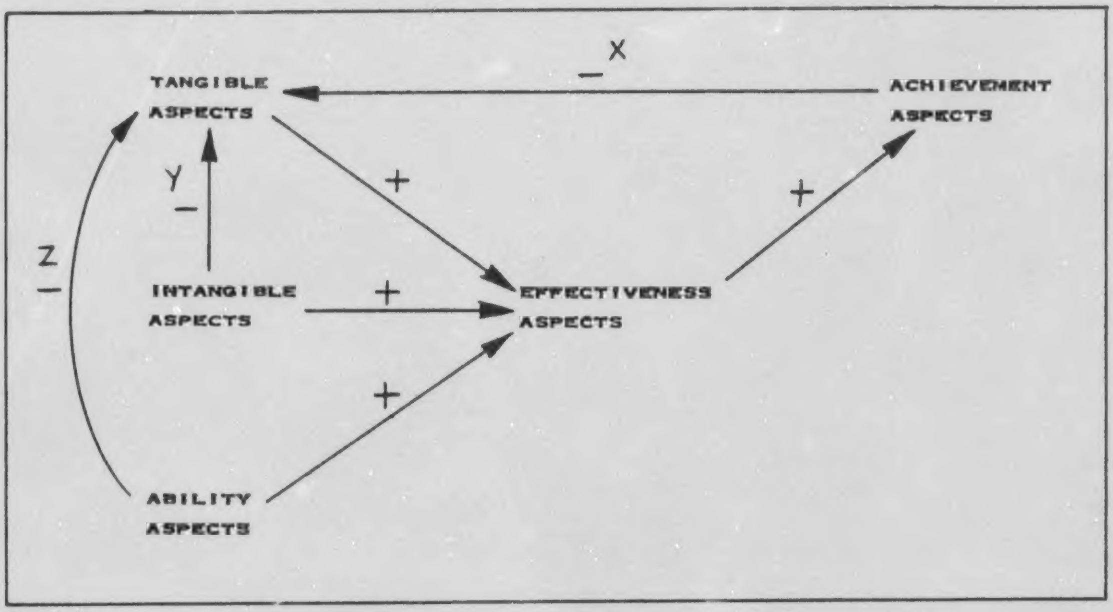


Diagram of the EEO Accountability Model

The arrows in the diagram show direction of impact of various "aspects" (or groups of EEO variables in this study) on one another. Thus, in the EEO Accountability Model, Tangible Aspects, such as expenditures per student and allocations for students, were seen to have a positive impact on effectiveness aspects, such as test score gains. Similarly, intangible aspects, such as teacher expectations and school racial balance; and ability aspects, such as I.Q. and reading ability, positively impact effectiveness variables. And effectiveness variables positively impact achievement aspects, such as grades and achievement test scores. By "positively impact" the model implies that an increase (or decrease) in the values of one set of variables is accompanied by a corresponding increase (or decrease) in the second set of variables. The more interesting relationships seen in the Accountability Model are the "feedback loops" formed by the negative impact of intangible aspects, ability aspects, and achievement aspects on the tangible aspects. These feedback loops ensure that the school system allocates its resources (tangible resources) proportionately where the need is greatest. These feedback relations define the accountability of the system for ensuring equality of educational opportunity.

Phase 5: The Decision Support System

In the final phase of the research, the model was incorporated into a Decision Support System in such a way that the DSS was able to calcu- late a single value, termed an EEO Accountability Index, for a given set of data.

In the above diagram the values of the feedback relationships (used here to mean negative impact relationships) are labeled with the letters $\underline{x}$, $\underline{y}$, and $\underline{z}$. These values were defined as average Pearson correlation coefficients between variables in the different classifications of EEO aspects.

In order to calculate the EEO Accountability Index, the values of the coefficients $\underline{x}$, $\underline{y}$, and $\underline{z}$ had first to be "normalized" (1) to avoid a zero factor (since correlation coefficients run from -1 to +1) and (2) to account for the sign of the coefficient. Normalized coefficients were thus defined as follows:

$$x' = -x + 1$$
$$y' = -y + 1$$
$$z' = -z + 1$$

Values of $\underline{x'}$, $\underline{y'}$, and $\underline{z'}$ defined in this way range from .01 to 1.99 for any set of data from a given school system. The stronger the feedback relationship, the higher the value of the coefficient calculated by summing some factor of each of the three normalized feedback coefficients. This summation defines the EEO Accountability Index, as follows:

$$\text{EEO Accountability Index} = ax' + by' + cz'$$

where $\underline{a}$, $\underline{b}$, and $\underline{c}$ are weighting factors and $\underline{x'}$, $\underline{y'}$, and $\underline{z'}$ are normalized average values of the feedback coefficients.

APPLICATION

A simple example of the use of the suggested EEO Accountability Index is in the choice between two alternatives. Let us look at the choice a school district may have between implementing an effective "dropout" program or an effective mathematics program where, due to budget restrictions, both programs cannot be implemented at the same time. For each of those two alternatives, the computerized decision support system would make the appropriate changes in the tangible (and intangible) aspect variables of the students to be affected, correlation coefficients and normalized coefficients would be calculated, and weighting factors to reflect values of the decision makers would be assigned. Then an EEO accountability index would be calculated and reported. Thus, there would be an accountability index, measuring the overall system EEO accountability and reflecting the values of the decision makers, for each alternative considered. The alternative with the higher EEO Accountability Index would best meet the overall EEO goals of the school system, within the context of the given value system.

It is most useful to find a single index to accomplish such a feat. And yet, the choice between two alternatives is perhaps the least impressive potential use of the suggested EEO Accountability Index. One can imagine a school district contemplating perhaps dozens of budgets, each reflecting different priorities and approaches. For each contemplated budget, a systemwide accountability index could be calculated. School district planning/budgeting functions could thus become more comprehensive, drawn together by the unifying goal of EEO accountability.

Within the Portland Public School District, just such an application of the suggested EEO accountability index is being undertaken. The district has been examining a number of proposals as to how staff should be allocated to individual schools. Some proposals call for increasing teacher support personnel, thereby reducing the burden on teachers to accomplish many different tasks. Other proposals call for allocation of more teachers (and therefore fewer support personnel, since total numbers of staff are limited by budget restrictions), thus reducing class size and the burden on teachers to handle more students. A number of proposals have varying degrees of teaching and support staff allocations. It is possible to calculate the allocation to each school under each proposal. With the application of the DSS and the suggested EEO Accountability Index, it is possible to calculate an EEO Accountability Index for each proposal. Values of the School Board, of the administration, of teachers, and of any other group, can be incorporated into the calculation of the indices by adjustments to the weighting factors. Thus, EEO accountability indices can be calculated for each proposal for each identified value system. These indices can then be used to support a decision on how to allocate staff within the district.

CONCLUSION

This paper presents research conducted by the author in the development of a computer Decision Support System for determining equality of educational opportunity in a public school district. Constant interaction with a task group of key school district administrators, and with other interested personnel, as well as with the Data Processing and Evaluation Department staffs, led to the development and implementation of a sophisticated DSS for determining an EEO Accountability Index. The potential usefulness of such a DSS seems quite valuable at this time.

REFERENCES

[1] Sexton, Porter, A Computer Decision Support System to Assist in Providing Equality of Educational Opportunity, Portland, Oregon: Portland State University, 1983.

[2] Clemson, Barry, "Beyond Management Information Systems," Educational Admininstration Quarterly, Fall, 1978.

[3] Coleman, James, "The Concept of Equality of Edcucational Opportunity," Harvard Educational Review, May-June, 1967.

[4] Beshers, James M., "Models of the Educational Process: A Sociologist's Perspective," in Mosteller, Frederick, and Moynihan, Daniel, P., On Equality of Educational Opportunity, New York; Random House, 1972.

TEACHERS' INSTRUCTIONAL USES OF MICROCOMPUTERS*

Richard Shavelson, John Winkler, Cathleen Stasz, and Abby Robyn

The Rand Corporation
17[illegible] Main Street
Santa Monica, California 90406

This paper is a progress report on a study of teachers' instructional uses of microcomputers, grades 3-12. We describe the study and some preliminary findings to date.

INTRODUCTION

The number of microcomputers in public schools increased a whopping 230 percent between fall 19[illegible]0 and spring 1982 (NCES, 1982). Despite this seemingly impressive number, and despite the ballyhoo that is made about computer-assisted education, there is much less to this number than meets the eye. Although roughly 100,000 microcomputers can be found in our public schools, this number translates into about 1 microcomputer for every 4 schools, or 1 micro for every 20 classrooms, or 1 micro for every 450 students. From these numbers, it is easy to see why most classrooms do not have ready access to micros; why there is insufficient time for most students to become literate beyond superficial operational characteristics of the machines; and why there is insufficent time for most students to receive computer-assisted-instruction (CAI) in any concentrated amount.

Clearly, this paucity of microcomputers is a major obstacle to reaching national goals for computer literacy and CAI, but it is not the only one. Two other obstacles suggest that the shortfall of hardware might be a blessing in disguise. They are the lack of adequate courseware and the lack of teachers well-enough versed to use computers effectively in their instruction.

The goal of this study is to set forth guidelines for designing educational courseware that meet teachers' needs and for educating preservice and inservice teachers in the instructional uses of microcomputers. To this end, the study examines the relationships among teachers' attitudes toward computers, their knowledge of computers and the subject matter taught, and their uses of microcomputers for instruction. This paper describes the study and reports preliminary impressions and findings. (Data collection is currently underway and will be completed by conference time.)

DESIGN OF THE STUDY

We posit that school-district policies toward the implementation and support for instructional uses of microcomputers along with the characteristics of the community and students served (e.g., socioeconomic status) will influence how many micros will be used in instruction directly and also indirectly by their influence on school administrators and teachers. Of particular importance is the influence of certain teacher characteristics--attitudes and knowledge--on their decisions for instructional uses of micros. In this study, district- and school-policies, and the characteristics of students serve as the context that moderates the focus of the study, teachers' uses of micros for instruction.

We planned to sample five school districts in California that systematically varied in microcomputer implementation and support policies, and student population served. Within each of these

---

*This study is jointly funded by the National Institute of Education and The Rand Corporation. The opinions expressed herein do not necessarily reflect the position or policy of the NIE or The Rand Corporation. The findings are preliminary, tentative and may be modified in the final report of the study.

five districts, two teachers who were identified as unusually successful in using micros for instruction were to be selected in each of two schools at the elementary, junior- and senior-high levels. All told, we expected to interview and observe 60 teachers (5 x 2 x 2 x 3), 30 principals (5 x 1 x 2 x 3) and 5 staff responsible for microcomputers in their districts. While the design was straight-forward, someone forgot to set up the world of education so we could implement it. We found that rarely, if ever, would two unusually successful computer-using-teachers be nominated in one school, let alone reproduce that phenomena in more than one school at the same grade level (e.g., elementary); let alone reproduce the phenomenon at more than one grade level (e.g., elementary and junior high); and so on.

The fact is, at least in the 43 computer-using districts in California that we have already contacted, the burden of meeting national goals of computer literacy and CAI is placed squarely on the shoulders of a very small, overworked cadre of teachers--"computer buffs"--in a school district. (For cost reasons, the study was limited geographically to California.) Our sampling plan, then, was modified such that we searched the state seeking nominations for these teachers. In order to find 60 such teachers, we will probably visit 40-50 schools and 10-15 districts.

SELECTED IMPRESSIONS AND PRELIMINARY FINDINGS

Variation in District Implementation and Support Policies. We found that districts vary greatly with respect to implementing microcomputers. Perhaps the most salient variation occurs in centralized coordination, which appears to be unrelated to district size or wealth. In one small high-school district, for example, policies are carried out by a central computer committee comprised of knowledgeable district staff and teachers and/or administrators from each school. Decisions about the purchase and kind of microcomputer, selection of courseware, provision of staff development and the like are coordinated by the committee. Equity across high schools in numbers of computers, availability of courseware and staff development, and use of micros across subject matters is a major goal of the committee. Needless to say, we had little trouble finding this district and once found, had little difficulty getting consistent nominations of successful teachers.

Contrast this with a large unified school district that has given lip-service to the importance of micros but has not formulated policies regarding implementation and support, and has not provided leadership in coordinating individual schools' efforts to provide literacy or instruction with micros. The chaotic state of the district is reflected in the fact that district personnel consistently referred us to other district personnel who, in the end, knew anything about computer use, least of all what their colleagues knew. We suspect that these kinds of policy differences will ultimately affect the instructional uses of microcomputers. But how centralized coordination helps and hinders teachers remains to be seen.

Variation in School-building Policies. Principals play a key role in creating effective schools. We were particularly interested, then, in how principals' policies for implementing and supporting computers might affect teachers' instructional uses of them.

We have found in interviews with principals, so far, little variability among their implementation strategies. Perhaps the three most striking findings so far are these. First, almost all principals favor instructional use of microcomputers. From what meager discretionary funds they have at their disposal (e.g., $4,000), they allocate or support departmental allocation of a respectable percentage to computer goals (e.g., purchase of hardware, courseware, or staff development). They give teachers release time to attend computer workshops or conferences. And they encourage teachers' interests and computer-related activities. Second, they place much of the responsibility for expanding the school computer program on the teachers. They do not coerce teachers into using micros but instead provide subtle encouragement by example. They delegate responsibility to committed teachers or to someone in the school's central office (e.g., a vice principal) to manage the details of their computer implementation policies.

What is remarkable is that all of this occurs in the face of considerable lack of knowledge about microcomputer hardware, courseware, or training alternatives on the part of principals. Very few principals owned a computer or knew how to use one for instruction, administrative tasks, or even recreation, despite professed interest in and commitment to computer literacy. What does this indicate? Perhaps it says something about effective management style--that is, positive attitudes and material encouragement may be more necessary at this level than exhaustive knowledge.

Variations in Teachers' Instructional Uses of Micros. In order to determine how various factors influence teachers' instructional uses of micros, it is necessary to develop a working idea of what constitutes "successful use." For this, we turned to the educational technology literature and found little to offer that was systematic or that considered how use of micros in instruction might be "successful" from the *teacher's* perspective.

For a definition of successful computer use, then, we turned to research on teaching and found the teacher decisionmaking perspective to be most useful. Teaching is viewed as a planning process, in which teachers integrate information about students, the subject matter, and the classroom and school environment in order to reach decisions about instructional activities. Goals, content, activities, and methods are orchestrated to maintain a flow of activity. Furthermore, teachers monitor ongoing activities and proceed as planned unless something interrupts the flow of activity.

To define successful computer use within this theoretical perspective, we first assume that classroom computer use fits *within* teachers' ongoing planning and decisionmaking processes. Next, we assume that teachers can make reasonable choices among alternative courseware for reaching one or some combination of goals, and among the modes of instruction (e.g., drill and practice, simulation) given their knowledge of the subject-matter, computers, and the characteristics of students in their classes. Successful computer use will arise when teachers make reasonable decisions about matching the computer and available courseware to their instructional goals, the subject-matter structure, the nature of students, and the instructional context. Nevertheless, once the planning decisions have been made, the teacher must possess the interactive teaching *skills* in order to carry out the plan.

This conceptualization leads to a definition of successful use of micros for instruction as the *appropriate integration* of computer-based *activities* with *teachers' instructional goals* and with *ongoing instructional activities*, which changes and improves on the basis of *feedback* that indicates whether goals are achieved. Instructional goals include *achievement* (mastery of basic skills and concepts) and *motivation* (time on task, interest). Ongoing instructional activities include *subject matter*, *concepts*, and *teaching aids* (texts; labs). Computer-based activities involve the *types of CAI* used (e.g., drill and practice, simulation), the *grouping of students* for computer activities (e.g., by size, ability), and the *selection of courseware* along some criteria. The appropriateness of integration involves both the *breadth* of computer-based activities, as well as the *mix* of selectiveness decided on (e.g., differentiations among computer activities for types of students or goals). Finally, feedback refers to the monitoring of processes and outcomes to revise uses of computers in instruction.

Thus far, we have observed considerable variations in instructional uses of micros. Until the full data set is systematically analyzed, consistent patterns are difficult to discern. Nevertheless, a few important impressions have emerged that provide a flavor of the findings.

The physical location of the micros plays a large role in how they are used instructionally, what teachers need to know in order to use them, how well computer activities are integrated with the regular curriculum, and how much time students spend working on them. First, elementary schools are much more likely to have one or two micros in the classroom while secondary schools are much more likely to have them in laboratories. Second, micros are more likely to be integrated into ongoing instruction if they are in or directly accessible to the classroom. That is, the ease of access to micros plays an important role in the degree to which they become coordinated with classroom instruction. There are, of course, exceptions, but what is most intriguing about this observation is that often districts with centralized planning find lab-type arrangements easiest to implement. This, however, may be counterproductive if integration into the curriculum is desired.

Most of the CAI we have seen used is drill and practice; we have observed fewer tutorials and simulations. The amount of time that students spend on the computer varies within and between classes. The difficulty of the activities for students also varies greatly; some move through the material smoothly while others make repeated errors. These variabilities may be due, in part, to the availability of courseware and terminals, to the quality of the available courseware, and to the overlap between the computer activities and the regular curriculum. These variations are generally not deliberate. Successful teachers, however, seem to be set apart by their attempt to individualize computer activities for students and to differentiate among different computer activities for different goals. These strategies, in turn, require some knowledge of possibilities for computer activities and the ability to select courseware and match it to students.

THIS WAY TO THE BAR...

Sally A. Sloan
Minneapolis Public Schools
807 N.E. Broadway Avenue
Minneapolis MN 55413

ABSTRACT:

A new "frontier in educational computing can be found in an input device that is beginning to be seen outside of education: the bar code reader. In Minneapolis, we have begun to use bar code readers in association with micro-computers (APPLE) and mainframes for a variety of purposes from inventory of learning materials to testing and even a lottery. A model program has been in operation since Fall '82 which has revolutionized the distribution and accountability of more than 75,000 items of instructional materials. Developments are underway regarding library operations as well as consideration of testing programs and attendance. This "new" technology may have serious potential for use in education.

When someone asks you the location of the nearest bar, do you assume they are looking for the local dispenser of alcoholic beverages or the location of the bar code labels for inventory purposes? While the former is not unknown in the environment in which we work, we are finding an increase in the occasions for the latter.

BAR CODE technology has been around for a very long time. While it has been used primarily in industry for inventory purposes, we are seeing its adoption in many aspects of daily life. Grocery stores have adapted bar code readers to facilitate check-out. Large companies, such as Sears, have adopted the technology to speed sales. Where once bar codes were found only in factories and storage sheds, we now see them as a part of most peoples' lives.

Like programming languages, there are several types of bar codes available. Again, like programming languages, they seem to be specific to certain purposes. The bar codes seen on groceries are not the same codes as those usually used for production inventories. The seven most common codes vary in length and characteristics, but all function in essentially the same way. The lines and spaces that are "read" make up the code. The first and last lines are signals of beginning and end of message.

MPLS PUB SCH - PIC
100-SLOAN SALLY

The bar code pictured here is called Code 39. It permits alpha as well as numeric representation. Since the coded message also appears in print above the bar codes it can be manually recorded in the event of power failure. This is the code type used now in Minneapolis.

Bar code readers vary with purpose. As you might expect they also vary considerably in price. The readers most commonly seen is stores may even make use of holography to permit accurate readings of cylindrical cans, etc. These are generally a "window" in the check-out counter over which the clerk passes the object. You may have noticed that it occasionally requires more than one pass for the object to be read. Most bar code readers emit a beep when a code has been successfully read and no tone which it has NOT been successfully read. Some readers have a complete set of tones and single or double beeps as indicators. Another commonly seen device is a wand type reader. The Sears stores use one shaped like a gun for easy handling. A simple wand shaped like a pencil is very common. The tips of bar code reading wands are some sort of light device. The manufacturers vie with each other for better tips to provide faster, more accurate, "first pass" readings.

Bar code readers can be built in to devices which insure that the code will be held the proper distance from the reader to insure correct "first pass" reading. When operating the hand held devices, operators need to learn the technique of reading codes. The wand must pass at essentially right angles to the code, be within certain tolerated distance of the code, and travel at an appropriate speed. The technology is such that a great deal of tolerance exists so that a naive user may expect to be able to operate a bar code reader almost immediately.

Bar code readers are available in portable, battery operated models as well as models which are "wired" into the host computer (as they are in grocery stores). The advantage of carrying a portable reader and collecting data in a factory storeroom situation is obvious. The same advantage occurs in the educational environment. These portable units generally allow very simple "downloading" to a host computer and contain more than adequate ammounts of storage. Some are even programmable to allow for prompts to the user.

CODE 39 TO MINNEAPOLIS Minneapolis Public Schools became involved with bar coding by way of a problem desperately in need of a solution. The Prescriptive Instruction Center (PIC) houses over 77,000 items of learning materials which are distributed among more that 450 Special Ed teachers. Inventory was a serious problem. Check-in and check-out were extremely time and labor intensive. In addition, Special Ed is required by law to submit extensive reports on the use of such materials.

Carole Gupton and Sue Kranz, of the PIC Center, perceived the need to computerize their operation and approached the Instructional Computer coordinator (author). The first step was to have all the materials recorded in an appropriate inventory program on a main frame computer. (CDC Cyber System owned by MECC from which Minneapolis schools buy time.) A Minneapolis teacher, Paul Dillenberger, wrote the software and assisted the PIC staff in the enormous task of inputting the inventory codes into his program. Because of restrictions on time, this task was completed primarily after school hours over a period of several months. The inventory code designed for the materials included classification code (subject matter), cutter code (type of material, e.g., books, equipment), copy number (e.g., which of three copies of nonconsumable) and shelf location. Teacher names, locations, and check-in/out dates were input as transactions occurred. Reports were now much more easily produced. There still remained a serious problem in check-out/in of materials. Teachers would spend considerable time filling out paper forms. Staff would then get on-line after hours and transcribe the information into the computer. Another solution had to be found.

An intrepid crew consisting of Sue Kranz and Carole Gupton from PIC, Paul Dillenberger, the MECC Coordinator Marcia Horn and the author visited a site where a bar code reader (wand type) was being used in a library. The technology seemed to hold promise for the task at hand. This was pursued by inviting vendors to come in and demonstrate their equipment. The notion of using a micro with a bar code reader seemed perfect for the environment.

The decision was made to purchase an Inter-Mec ruby wand scanner and an Inter-Mec bar code printer for the PIC. While it is possible to print acceptable bar codes on an Epsom printer using an APPLE, the quality, ease of application, and durability of the labels printed on the Inter-Mec printer, as well as the likely district-wide need for bar code labels in the future, made the purchase of the special printer cost-effective.

A portable Inter-Mec reader was purchased for the Computer Department to assist PIC when necessary and to be used for training and in exploring the role of bar code technology in our environment. It has 66k of memory, is programmable, and interfaces with an APPLE microcomputer.

The PIC inventory codes were downloaded from the mainframe in order to print the bar code lables without further need for typing. An APPLE was employed to drive the bar code reader. Currently both the wand and the printer are attached to an APPLE. Teacher information has been bar coded and filed on a Roll-A-Dex. To check-out/in materials, the codes on the materials are read, the teacher information is read off the Roll-A-Dex and the transaction recorded on APPLE diskettes. The daily transactions are uploaded to the mainframe during low usage hours on the time sharing computer. The time involved for staff and teachers is significantly shorter now that bar coding has been implemented.

When investigating the possibilities of using bar codes, the first question was how much programming would be necessary to implement its use. We were pleasantly surprised to find that the Inter-Mec required no special programming. It connects to a card in slot 4 on an APPLE. To input merely requires IN#4 imbedded in the driving software or from the keyboard. The way Dillinberger wrote the inventory software, staff has the option of inputting directly from the keyboard or using the bar code reader.

While there was no special software needed to use the bar code reader, the same cannot be said for the overall program. Dillinberger's program has the following capabilities:

Inventory: print inventory list, add or delete items from the list, print inventory bar code label, print list of consumables on hand.

Circulation: check-out/in an item, track item in circulation.

Retrieval: print retrieval list, user list, add/delete/change user information, print mailing labels of the list.

Usage Reports: print list of users by program, identifying number of instructional materials (nonconsumable/consumable), equipment and assessment tolls used, print program report identifying number of users, number of instructional materials (consumable/nonconsumable), equipment and assessment and tools used.

This program has been so effective it is under consideration for reproduction in a generalized form that could be made available to other districts. Future plans for PIC include automation of the book catalog with access points of curriculum objectives and learner characteristics.

As with all major computer applications, it took considerable time to implement this extremely successful operation. The first year involved long hours of typing in data. The second summer was spent attaching bar code labels to all materials. That task had to be done with great care to insure that the correct code was placed on each item. It resulted in a side benefit in that a positive inventory was then taken of all materials and all materials were certainly in their correct shelf space. If no material existed for the bar code label, that item was clearly missing. Now that all check-out/in is automated, the inventory is always correct and immediately available.

The Minneapolis School District is exploring the use of bar code technology for all equipment inventory. We are constantly in search of satisfactory software for media center check-in/out/overdue and inventory. There is even a pilot, in planning stage, to investigate the use of bar codes as part of an automated attendance package. (Rather than scan cards or sheets, the attendance clerk would use a bar code reader to input student and staff information.) What the future will bring in the way of new uses of bar codes in the education environment is restricted only be the degree of imagination and creativity of staff.

The Computer - A Greater Role in Accounting Education?

Steven W. Smalt, M. Acc.
Instructor of Accounting
School of Business
Columbus College
Columbus, Georgia 31907

and

Rodger Trigg, Ph.D.
Associate Professor of Accounting
School of Business
Columbus College
Columbus, Georgia 31907

This study explores the effects of ever advancing computer technology on accounting education. There is an obvious need for accountants and auditors to gain greater expertise in this area particularly given the prevalent use of computers in business. In addition to developing this need, the paper examines the results of a survey performed by the authors from accounting departments of 200 AACSB Schools of Business.

Formal recognition of the need for computer education in accounting curricula came about in the early 1970's. After 10 years, 70% of the 93 accounting departments responding have not used the computer as a teaching device in accounting education.

The paper explicates some of the methods that could appropriately assist accounting educators in meeting this challenge.

Introduction and Background

The days of the manual accounting systems are quickly drawing to a close. Accounting educators can boast of quality output only if former students are adequately prepared for what lies ahead in accounting practice.

The accounting profession, out of necessity, has had to adapt to an ever advancing technological environment. The computer has made it possible to record and account for the countless number of transactions occuring in business annually. The profession's dependency upon the computer can only increase, particularly as computing costs continue to decline. Computerized accounting systems can now be justified (cost/benefit) by virtually every business, from "Mom and Pop" to the multinational multi-billion dollar conglomerate.

This increased dependence upon technology has placed critical demands upon accounting education. However, the needs of this changing accounting environment have not gone unpredicted. The committee on Accounting Education, American Accounting Association and the Computer Education Subcommittee of the American Institute of Certified Public Accountants reported in 1974 that accounting graduates would almost assuredly become involved with the computer whether in the practice of public accounting or private accounting. The primary concern of this committee was that the accountant not be awe struck at the thought of the computer.[1] Accountants should be able to distinguish the advantages and disadvantages of computerized accounting. They should be able to recognize the computer as a tool with seemingly limitless applications to assist them in the performance of necessary accounting tasks.

Survey

The purpose of this paper is to report the results of a survey designed to evaluate the progress being made by accounting education in light of the above. Accounting departments from two hundred college and university Schools of Business were surveyed. Useable responses were received from 93 participants, for a total useable response rate of 46.5%.

Need Confirmed

As might have been expected when asked about the need to increase accounting student's knowledge about computers, 87% of those responding agreed there is a need in this area. Sixty-five percent of those agreeing strongly agreed with the proposition that this knowledge be fostered in accounting education.

The questionnaire also asked the accounting educators to rank nine minor areas of study, the first being the most useful and the ninth being the least useful. Table I below depicts the percentage of each ranking relative to computer science (information systems) as a minor area of study. The other minor areas of study ranked by the respondents included management, marketing, psychology, economics, statistics, quantitative methods, finance, insurance/real estate, and any other area desired by the respondent.

TABLE I

Computer Science
(Information Systems)
Minor Rank Percentages
(1-Most Useful 9-Least Useful)

Rank	Percent
1	49%
2	11%
3	13%
4	4%
5	12%
6	6%
7	2%
8	1%
9	0%
	100%

Table I illustrates that 73% of those responding chose computer science (information systems) among their top three choices as useful minor areas of study. The accounting educators responding apparently sense the necessity to increase skills in this area for success in accounting practice.

Using the Computer?

Sensing the increased need for a computer interface in accounting education and actually using the computer certainly appear to be two different things. The

310

accounting departments responding indicated that 70% had not utilized the computer as a teaching device in accounting courses. Specific courses in Accounting Information Systems are being offered in increasing numbers by accounting departments to assist in the accounting/computer interface however 70% of the respondents do not currently have accounting course work in this area.

Although a large percentage of respondents are not currently incorporating a computer interface in their accounting coursework, 61% indicated that they would use the computer if a feasible method of presentation existed.

Increasing Computer Usage

The importance of an accounting/computer interface has been established. How to convey relevant computer knowledge to accounting students is the critical question, particularly given the respondents willingness to utilize the computer in a teaching environment.

The survey gave accounting departments an opportunity to express how in their opinion the difficulty of presentation could be resolved. Table II illustrates that 48% of those responding favor the use of practice sets that utilize the computer. In the past practice sets that offer this interface function rather than being accounting oriented. Several publishers have recently begun to market practice sets that not only educate students regarding the functions performed by the computer, but also provide instruction in accounting concepts. Table II also indicates that 35% of the respondents suggested revising accounting texts to incorporate computer topics relevent to accounting. This approach has also been recently taken by numerous publishers. Accounting texts at all levels are beginning to discuss computer topics. The major difficulty with utilizing this approach alone is that students only become exposed to technology from a text rather than "hands on" experiences. The use of supplemental texts, guest instructors and working special problems all have the same weakness as the revision of accounting texts. This is not to say that these approaches to solving the accounting/computer interface problems should not be utilized. A combination of revising texts and utilizing computerized practice sets will probably provide the optimum utility.

TABLE II

Percentage of Respondents Suggesting Each Approach

Approach	%
Supplemental Texts	8%
Accounting Practice Sets Utilizing the Computer	48%
Revise Accounting Texts to Incorporate Computer Topice Relevant to Accounting	35%
Guest Instructors	2%
Special Problems	3%
Other Suggestions	4%

Conclusions

As indicated above a majority of the respondents stated that they would use the computer if a feasible approach were available. Feasibility, however, does not merely mean the availability of a computerized practice set and revised accounting text. In a survey performed in 1974 lack of qualified faculty, lack of belief in the computer as an effective instructional tool and lack of funds were the three major areas inhibiting the use of the computer in accounting education.[2] The apparent willingness of accounting educators to utilize the computer as an instructional device would indicate a possible change in thinking over the past nine or ten years about the computer's potential effectiveness. The overall affordability of computers has surely had an impact upon the number of institutions with computing facilities available both for administration and instructional purposes. The final major obstacle in 1974 was the lack of qualified faculty. The purpose of this study was not to evaluate the qualification of faculty but rather to evaluate accounting educators' suggestions for implementing the accounting/computer interface.

Progress has been made. However, the pace at which we as accounting educators are moving is painfully slow in relation to the technological advances being experienced in professional practice. If the fruits of our labor are to be prepared as they move from acedemic careers to professional careers we must see the need and supply the product to satisfy the need.

Footnotes

1. "Inclusion of EDP in an Undergraduate Auditing Curriculum", Journal of Accountancy, Volume 138, Number 6, December 1974, pp. 118-121.

2. Solomon, Lanny, "The Computer in Undergraduate Accounting Education: A Survey", Journal of Accountancy, Volume 138, Number 6, December 1974, pp. 121-125.

A COMPUTER SCIENCE CURRICULUM THAT ISN'T THE SAME

Lyle B. Smith
Computer Science Department
Northern Illinois University
DeKalb, IL 60115

ABSTRACT

There are many colleges and universities that have programs and grant degrees called Computer Science but the content of the programs can vary widely. The NIU Computer Science degree was designed some twelve years ago to produce a truly professional data processing person. Over the years the content has evolved with the computing field so that many of the NIU Computer Science graduates are hired at a level equivalent to other people with one or two years experience after their degree. Careful selection of homework topics, a heavy load of programming assignments, and attention to state-of-the-art needs and practices in business are some of the factors that lead to making NIU graduates different.

INDUSTRY NEEDS "PROFESSIONALS"

Webster's Collegiate Dictionary says (in our context): "Professional,n.: one that engages in a pursuit or activity professionally" and "Profesionally, adv.: following a line of conduct as though it were a profession, or, characterized by or conforming to the technical or ethical standards of a profession" and "Profession, n.: a calling requiring specialized knowledge and often long and intensive academic preparation." When you hire an attorney you assume you are employing a professional. . .that they know their business and how to get results. When you think about medical doctors you assume they are professionals. . .that they know medicine, have the proper equipment and knowledge to diagnose and cure.

Industry wants programmers who are no less professionals than doctors or lawyers! They need programmers (and systems analysts) who know their business and have the correct tools and know-how to get jobs done efficiently and correctly.

Computer "systems" that include all the central and peripheral hardware, operating systems, teleprocessing systems, data base management systems, compilers, automated file handling, and a library full of programmed utility processes are complex indeed and it takes a professional to handle them.

Some of the largest expenses to industry in the data processing area over the last two decades have been in maintaining and rewriting programs. Often well over 50% of a company's programming work has been on maintenance.

Programming done by a professional is well written (logically and correctly), well documented and well tested. It can be fixed or modified by someone else.

Programming as a profession has no life of its own! A program must do something for someone. The point is that any program has its own environment: a computer room, a classroom, a business, or you-name-it. In order to do a professional job the programmer must be aware of, understand, and interact with that environment.

Industry wants professionals. People who:

1. Know how to make the computer perform specific tasks. They know how to write a program and how to get that program to run.
2. Know a variety of ways to do things so that they can be confident of selecting an efficient method.
3. Have an understanding of the entire "system" so their programming solution avoids problems and is adaptable to future changes as appropriate.
4. Are knowledgeable in the most recent technological developments so that they can utilize new techniques to advantage.
5. Do professional work. Their programs are well structured, logically sound, amply tested, well documented, they work efficiently and they can be modified by someone else with ease.
6. Know the environment and can communicate effectively with others in that environment. This characteristic of a professional must not be underestimated. The professional programmer is providing services (in the form of production programs) to others whose needs must be understood if they are to be met.

When W. E. Singletary moved to NIU twelve years ago to build the computer science program, he had a vision. What he saw was a large industrial area (Chicago) that would need many professional programmers. He saw representatives of the industries interacting with the faculty to articulate industry's needs. Furthermore, once the program was underway, the industry-educator conversations would continue so that as computing technology changed the curriculum could keep pace. Now, let us see how Singletary's vision became a reality.

MEETING THE NEED FOR PROFESSIONALS

There are many, many public, private and industrial programs to teach computing, computer science, data processing or whatever you prefer to call it. No one department can be all things to all people. Therefore an initial decision was to carefully define the scope of the professional program being developed.

The largest need for people is in the area called business applications - designing, writing, debugging and testing the programs that handle the information processing needs of a company. The second largest need is in the area called systems programming - maintaining and improving the operating systems and a library of utility programs that are the environment in which applications programs run.

A primary decision was made to create professionals in the large demand areas of business applications and systems programming.

Businesses use a variety of brand-name computers: UNIVAC's, Control Data's, IBM's, DEC's, Amdahl's, Honeywell's, etc. In maybe 8 of 10 larger businesses you find IBM computers.

Another decision was made to create professionals who learned on and understood IBM computers (their hardware, languages, utilities and operating systems). Experience has shown that other manufacturers' systems can be easily learned after the first one is mastered.

The areas of computing in which one could become professional were narrowed further through discussion with industry as to their needs. The languages most used to write business applications are COBOL and ASSEMBLER. The professional will be able to apply the principles and concepts they know to quickly acquiring a working knowledge of a new language.

"Know thy environment" we noted previously as a leading commandment for a professional programmer. NIU has a respected College of Business and industrial representatives readily agreed that inclusion of Business courses would satisfy the environment commandment for our professional in business applications. Courses in Management, Marketing, Accounting and Finance were made available by the College of Business.

"But he can't even write a complete sentence!" is not a remark we would expect about a professional. NIU's computer science majors are required to take a technical writing course taught by the English Department.

Mathematics does not play a large role in business computing. Therefore there is not an emphasis on mathematics requirements. Our professional need only take a two semester sequence in analysis or calculus.

Statistics does show up in business information processing frequently. One statistics course is required to insure that the professional programmer is familiar with the terminology, calculations and usefulness of statistics.

The remainder of the cirriculum decisions are a natural fall out of the ongoing discussions with industry about their needs. Following the obvious inclusion of COBOL and ASSEMBLER, we have courses that cover JCL (Job Control Language), Utility programs, more advanced COBOL, internal data structures with advanced ASSEMBLER, data base and data communications (using IBM's IMS and CICS as examples), application systems design, and a course covering the general concepts and some detail of computer (hardware) architecture and operating systems.

The curriculum as outlined may not seem broad--it is not intended to be all inclusive. It _is_ intended to extract those topics that a programming professional in a business environment must be good at.

This program does not require many credit hours (about 43 from the 121 required for a Bachelor's degree). This leaves a lot of room for an individual to take other courses and develop into the well-rounded person a professional should be.

There are two more areas that the NIU program emphasizes in response to industry's needs. One is _structure_, in programs and in system design. The other is _documentation_.

The following table gives a summary of this cirriculum that produces professionals in business applications computing.

A Professional Curriculum in Business Applications Computing

Computer Science Requirements

COBOL
ASSEMBLER
EXTERNAL DATA STRUCTURES (JCL, UTILITIES, ADVANCED COBOL TOPICS, ISAM)
INTERNAL DATA STRUCTURES (ADVANCED ASSEMBLER, INTERNAL STRUCTURES)
DATA BASE & DATA COMMUNICATIONS (IMS, CICS, GENERAL CONCEPTS)
SYSTEM DESIGN (STRUCTURED

ANALYSIS AND DESIGN OF APPLICATION SYSTEMS)
COMPUTER ORGANIZATION (HARDWARE ARCHITECTURE AND OPERATING SYSTEMS)

Other Requirements

TECHNICAL WRITING (taught by English)
ANALYSIS OR CALCULUS (Two semesters)
STATISTICS (One semester)
ACCOUNTING Pick 3 of 4
FINANCE (taught by
MANAGEMENT College of
MARKETING Business)

80 (approx) additional credit hrs.

UNIVERSITY GENERAL EDUCATION REQUIREMENTS
ELECTIVES (for the rounded professional)

Overall Emphases

1. HARD WORK (many programs to write).
2. RELEVANT ASSIGNMENTS (programming assignments and carefully selected to teach business needs/practices).
3. STRUCTURE (programs, flowcharts and system designs are required to be STRUCTURED).
4. DOCUMENTATION (documentation standards for flowcharts, program comments and neatness of the program printout are set very high).

In the next section we will point out how the emphases on hard work, relevant assignments, structuring and documentation make what might otherwise be an average program into one that is different.

HARD WORK ETC. ARE THE KEYS

More than one coach of a team sport (e.g. football, soccer, etc.) has produced a winning team with the following four rules:

1. Learn the basic skills
2. Conditioning
3. Conditioning
4. Conditioning

Even though athletics and academics may seem to be at odds, these four rules do merit some attention. As they apply to Computer Science or to producing a computing professional they are:

1. Learn the basic skills
2. Programming
3. Programming
4. Programming

The COBOL and ASSEMBLER courses at NIU vie for top billing as "the hardest course I've ever taken." The intellectual concepts are not what makes the courses hard--it is the sheer work (conditioning) of programming and debugging and documenting the many non-trivial programs that are required.

Students considering a Computer Science major at NIU are advised that "it's a lot of hard work" and even so we cannot accept all those who apply.

However, just hard work is not the whole answer. The professional in computing must program "real" problems.

Homework programs typically involve 1000 to 1500 lines of COBOL and include the basic concepts that are required for almost all applications. The programming assignments in all the courses are selected to be samples of the type of program that is written in industry.

Standards for structured programming are now well defined and strictly adhered to. As the concepts become clearly defined, obviously useful and desired by industry they were included.

Documentation has been the "joke" of the computing industry for many years. Documentation standards for homework programs at NIU are well defined and strictly enforced.

In summary, the keys to success in developing a professional programmer are to apply the following four rules to a curriculum that addresses needs:

1. Learn the basic skills (include structure and documentation standards).
2. Programming (or relevant problems).
3. Programming (or relevant problems).
4. Programming (or relevant problems).

Does it Work?

How do you measure the success of an academic program? We suggest that you ask those who employ the graduates. Employers indicate their opinion of a prospective employee in two related ways. First by making a job offer or not and if so, by the level of salary offered.

Companies (many of whom are involved in continuing cirriculum discussions with the department) who recruit Computer Science professionals at NIU come back year after year. The reputation of the department has begun to spread in recent years and in addition to Chicago area recruiters, we have companies from both coasts who come to NIU for recruiting.

Students too can give a measure of success to a program by the number of them who wish to declare a Computer Science major. In recent years the number of students has surpassed the department ability to provide classes. A variety of restrictions (such as a GPA requirement) have had to be instituted to control the number of majors to match the department's resources.

The curriculum does change to reflect industry needs. The inclusion of structured programming and design was mentioned previously. The course that includes teleprocessing and CICS was previously optional and at industry request is now required.

So, is it a success? In the fall of 1981 the media picked up on a (true) story that one student at NIU had offered another $300 cash to give them their position in a computer science class that was already filled. Somebody thinks the courses are worth taking. However, ask the employers if these people are true professionals.

CG101 - A First Course in Computer Graphics

Lyle B. Smith
Computer Science Department
Northern Illinois University
DeKalb, IL 60115

ABSTRACT

Computer Graphics is fun, satisfying, and now, with personal computers, available. No longer is graphics only for those who can afford expensive equipment. No longer should there be any mystique about computer graphics. A first course in computer graphics is designed for first or second year college students with a working knowledge of BASIC or FORTRAN and access to something like an Apple or a large system with a more or less standard graphics subroutine package and plotter, COM, and/or CRT suitable for graphics output. The course includes 1) non-technical discussion of the various hardware, software and systems available for computer graphics, 2) specific examples of doing computer graphics and 3) a variety of programming assignments ranging from use of the basic graphics primitives to more sophisticated exercises that lead to understanding the finer aspects of CG such as hidden lines, shading, and the rotation of solid representations.

INTRODUCTION

The purpose of this paper is to outline a first course in Computer Graphics for students at or near their first year in college. Only a very basic level of mathematical background is assumed so as to make Computer Graphics available to all.

Generating pictures by computer is satisfying and fun. By appropriate selection of in-class examples and programming homework assignments, the fun and satisfaction can be conveyed to students of Computer Graphics.

A week-by-week outline of a fifteen week semester class (3 hours/week) is presented. Suggested programming assignments corresponding to the same fifteen week schedule are also given. The assignments are intended to incorporate the concepts learned up to that point in the semester. Modification of or substitution for the assignments are encouraged so as to maintain or promote additional student interest.

WEEK-BY-WEEK OUTLINE

Week 1: Introduction
- Define Computer Graphics (CG)
- Present a Historical Perspective
- Discuss the large variety of uses
- Show slides and/or video tapes with CG examples
- Homework - Reading

Week 2: Hardware
- Discuss CG Hardware

    - Vector graphics
    - Raster graphics
  - CRTs
  - Plotters
  - Input devices
    - Tablet, joystick, etc.
  - Show slides/video tapes
  - Give a live demonstration
  - Homework - Reading

Week 3: Software & Systems

- Adding graphics to a programming language
  - Graphics commands (BASIC)
  - Subroutine calls (FORTRAN)
- CG Systems
  - Hardware & Software working together
- Homework - Reading
  - Copy a CG program and execute it

Week 4: Graphics Primitives

- The six graphics primitives
  - START, ERASE, POINT, LINE, TEXT, STOP
- Illustrations of complex pictures built only from primitives
- Types of pictures
  - X-Y plots (graphs)
  - Line drawings ("wire figures")
  - Raster drawings
- Homework - Reading
  - Program 1, a picture using only primitives

Week 5: Advanced Software Tools

- Built-in (or programmed) functions
  - Automatic axes and scaling
  - Geometric figures (circles, etc.)
  - Curve plotting (a set of points)
- Picture subroutines
  - Repeating partial pictures
- Homework - Reading
  - Program 2, a relatively complex picture

Week 6: Interactive Graphics

- Define Interactive Graphics (of other levels)
  - Static
  - Interactive
  - Motion graphics
- Interactive Techniques
  - Menu driven (select an option)
  - Input driven (specify a value)
    - Numeric (keyboard)
    - Geometric (graphic device)
- Programming Techniques
  - Redraw entire picture
  - Redraw portions affected
- Homework - Reading
  - Program 3, simple interaction
    - a) a menu program
    - b) a coordinate input program

Week 7: Special Techniques

- Windowing
- Clipping - give clipping subroutine
- Changing Coordinate Systems
  - User coordinates
  - Device coordinates
- Homework - Reading
  - Program 3 (continued)

Week 8: Perspective and Steroscopic Projections

- Perspective views
  - Discuss the geometry and give a subroutine
- Stereoscopic views
  - Discuss the geometry and give a subroutine
- Homework - Program 4, draw a figure and
  - a) plot a perspective view
  - b) plot a stereoscopic projection
  - c) plot a clipped version

Week 9: Motion

- Movie frames
- Real time motion
- Homework - Program 5, Motion graphics
  - a) Draw several frames of a movie, or
  - b) Program motion on a CRT

Week 10: Generic Graphics

- Discuss device independent graphics
- A graphics file
- Develop graphics primitives for printer plotting
- Homework - Program 6, Printer Plots
  - a) Write printer plotting primitives
  - b) Use the primitives to create several plots

Week 11: Business Graphics

- Line Charts (give a subroutine)

Bar Charts (give a subroutine)
Pie Charts (give a subroutine)
Homework - Program 7, Business Graphics
  a) plot a line chart
  b) plot a bar chart
  c) plot a pie chart

Week 12: Complete Systems
CAD
CADD
CAD/CAM
Discuss vendor packaged systems and their capabilities
Homework - Reading
  Finish previous programs

Week 13: Other Graphics
Art
Design
Education
Etc.
Homework - Reading
  Program 8, Be artistic

Week 14: Game Graphics
Discuss near miss detection etc.
Discuss video game principles and how they apply to CAI
More on motion graphics
Homework - Reading
  Program 9, Program a simple "game" (optional)

Week 15: Everything Else (A preview)
Graphics data bases
Graphics Languages
Advanced Techniques (shading, color, etc.)

SUGGESTED PROGRAMS

1. A simple picture using only primitives (static)
   a) Draw a geometric figure (an n point star)
   b) Draw the floor plan for a house (room)
   c) Draw something artistic (using lines and points)

2. A more complex picture using automatic axis generation, scaling, circle generator, etc. (static)
   a) Plot a graph with several different curves
   b) Draw a stick figure holding a bunch of balloons

3. Simple interactive program (interactive)
   a) Write a program to accept coordinate input (points) and connect the points by lines - a drawing program.
   b) Write a program to display five different pictures selectable from a menu.

4. Use Clipping, Perspective and Stereoscopic (static/interactive)
   a) Draw a picture and show a perspective view, a stereoscopic view, and a clipped version of the picture.

5. Motion Graphics (static/interactive)
   a) Draw several (6) frames of a movie
   b) Program a moving object on a CRT

6. Printer Plotting (static)
   a) Write a set of printer plotting primitives
   b) Use the primitives to plot the pictures drawn for Program 1

7. Business Graphics (static/interactive)
   a) Write a program that uses the given line, bar or pie chart subroutines to plot the same data with each type of chart.
   b) Make the program interactive--accept several (5) input values, use a menu to choose a chart type, display the chart, and wait for the next choice from the menu.

8. Draw something artistic (static/interactive)
   a) Use your imagination

9. Game Graphics
   a) Write an interactive program with motion (a simple video game)

REFERENCES

Several texts on computer graphics and interactive techniques have been published. All appear to be quite mathematical so may not be entirely appropriate for this first course. Here we list a sample set. References (1) and (3) could provide reading material for this course, however they are quite mathematical. A more appropriate text is being written by this author.

(1) Angell, Ian O., A Practical Introduction to Computer Graphics, John Wiley & Sons, New York, 1981.

(2) Foley, James D. and Andries Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley, Reading, MA, 1981.

(3) Newman, William M., and Robert E. Sproull, Principles of Interactive Computer Graphics, 2nd Ed., McGraw-Hill, New York, 1979.

(4) Parslow, R. D., R. W. Prowse, and R. E. Gran, Eds., Computer Graphics, Techniques and Applications, Plenum Press, London, 1969.

Software Evaluation: Support Through the Texas Education Computer Cooperative (TECC) Project

Vicki S. Smith
Coordinator for Computer-Based Instruction
Region IV Education Service Center
P.O. Box 863
Houston, Texas 77001

The following describes a project funded through the Texas Education Agency for the purpose of increasing the effective use of microcomputers, specifically through courseware evaluation.

The emergence of microcomputer technology in the 1980's has demonstrated its ability to impact the effectiveness of both instruction and learning. The interactive nature of the microcomputer has helped it to gain wide acceptance among educators. As the cost of the hardware has decreased, schools have even more readily accepted the use of such a tool, however, often before proven courseware resources are available to support its use.

As the use of microcomputers expands in educational settings, there is evidence to support the fact that they assist the instructional process effectively only to the extent that quality courseware is available and accessed. There then becomes a great need for careful courseware evaluation, to determine what exactly is "good" and what is inadequate. The quality of courseware is improving, competition is becoming more intense, and greater amounts of money are being spent. The decisions to be made are growing more difficult, and schools must be able to justify such decisions.

Certainly a great responsibility will lie with the trained, computer literate teacher in how the hardware and software is used, but much of the success - or failure - of microcomputer use in the schools will depend upon the courseware selected.

The schools need a set of standards or guidelines whereby they might begin to determine what quality software is. Such guidelines not only assist them in purchasing, but also direct developers and evaluators. Schools need a coordinated effort to provide them such guidelines.

This need is being addressed by a project endorsed by the Texas Education Computer Cooperative. TECC, a group commissioned by the State Commissioner of Education, coordinates computer services in the multiregional and regional processing centers of the state. The group recommended that funds be granted for the creation of a developmental project to address issues related to microcomputer courseware. Acting on the recommendation of the TECC, the Texas Education Agency funded the TABS Related Courseware for Microcomputers Project. The project intent was to increase the effective use of microcomputers, especially as they assist instruction relating to the Texas Assessment of

Basic Skills (TABS), the State Board of Education adopted objectives, for reading and/or mathematics for grades 3, 5, and 9.

The Project Advisory Committee was comprised of one representative from each of the twenty Education Service Centers in Texas. Evaluative criteria and procedures were identified. Region IV Education Service Center, Houston, coordinated training for the twenty representatives, who in turn conducted training in evaluation techniques to approximately two hundred teachers across the State. These teachers were then responsible for the evaluation of each courseware package submitted.

Courseware designed for the Apple, Commodore PET and Radio Shack TRS-80 microcomputers was evaluated, and both commercial and non-proprietary sources were included. Region IV Education Service Center invited vendors to participate in the project by providing courseware packages for evaluation, then collected, catalogued, and disseminated courseware and evaluation guides to each of the twenty Education Service Centers.

These evaluation guides were developed following investigation of similiar efforts and instruments as MICROSIFT, CONDUIT, National Council of Teachers of Mathematics (NCTM), and EPIE. Courseware evaluations produced data in four general areas: identification, instructional strategies, relations to the Texas Assessment of Basic Skills (TABS), and specific instructional/technical qualities. Each courseware package was evaluated at two different sites for appropriate hardware, instructional design aspects, and correlation to specified skills.

Project work began in February, 1982, and by its conclusion in August, over two hundred courseware packages were evaluated. Data relating to identification were collected and catalogued by Region IV staff assigned to the project. Data addressing instructional strategies and TABS objectives were generated by the evaluation team acting as an entity. Finally, each member of the team individually rated the package on its specific instructional/technical qualities.

At the completion of the evaluation period, all materials were returned to Region IV, where evaluations were tabulated and final results disseminated. Results were compiled using three databases, one each for Apple, Radio Shack, and Commodore PET. These databases, although commercially developed, used formats specifically adapted for the project. Databases were published in both diskette and printed versions. Copies of both versions were disseminated to each Education Service Center in Texas for use by their respective school districts.

The TECC project is currently in its second phase. As such, the Statewide Microcomputer Courseware Evaluation Network has expanded its focus to include K-12 Reading/Language Arts, Mathemetics, Business Education, Vocational Education, and Science. Evaluation will largely parallel that of the first phase, although special interest will be given to those courseware packages which include a management system or which are generic tool packages (i.e., word processing, data base management, and similar programs). To insure effective evaluations, courseware packages will be evaluated at three sites across the state, and the evaluation form will see the inclusion of a student evaluation section. This phase of the project will be completed and the resulting evaluations available by September, 1983. Again, these will be distributed to the Education Service Centers for use by their school districts as a guide for courseware procurement.

Production of the database has, obviously been the primary project goal. However, other rewards or accomplishments have been realized as well. Educators in Texas are becoming more computer literate as they work with courseware evaluation techniques and databases.

Education Service Centers and local school districts are accomplishing much by working together. Vendors and software developers who receive copies of evaluations for their submitted packages are discussing ways they might better and more specifically address teacher needs.

The overall intent of the TECC project is not to recommend specific hardware or software, but rather to serve as a tool in the schools' decision-making process of previewing and procuring the appropriate software for instructional use. It is meant to provide information rather than to advocate specific methodology or technique. Hopefully, the product information will serve as a "consumer's guide" to reference, before they purchase, for determining whether the courseware is effective, whether it fits into the instructional and curricular patterns of the district, and whether it addresses specific objectives. The role of the TECC project is one of support.

As microcomputers proliferate in the schools from almost endless vendors, courseware assumes a more and more strategic position. Through the TECC courseware evaluation project, steps are being taken to ensure that microcomputer courseware will be appropriate to the tasks faced by Texas educators.

USING EXPLORATORY DATA ANALYSIS TO EVALUATE STUDENT PERFORMANCE IN THE CLASSROOM

by

H. Ray Souder
William Leigh
William Holloway

Northern Kentucky University
Highland Heights, Kentucky 41076
606-572-6357

ABSTRACT

This paper describes an alternative way to evaluate students in their classroom performance. Exploratory data analysis is a way of thinking about data analysis and also a way of doing it. The exploratory approach to data analysis seeks to have the user identify relationships among data that may go unnoticed with more traditional techniques.

Exploratory data analysis is an alternative method to evaluate a student's performance in the classroom. The major benefits in utilizing EDA are 1) the distribution better represents the typical student performance, 2) order statistics are more resistant to outliers, 3) the grade assignment process is more objective, and 4) the technique is intuitively understood by instructors and students alike.

A never ending problem of evaluating student performance confronts all teachers K through the post-secondary levels. Most instructors rely on traditional statistical techniques, for example, mean, variance, and standard deviation to accomplish this evaluation process. These traditional statistical methods are known as a confirmatory model of analysis. In this model of analysis, a model of the relationship is fit to the data; statistical summaries are derived and are tested against the probability the obtained values have occurred by chance. These tests of probability are derived by likelihood ratios. A confirmatory model is designed to address the question: "Do the data I have confirm a given hypothesis?"

An alternative approach is to address a different question: "What can the data reveal to me about the relationship between the variables observed?" This question allows a wide range of alternative explanations to be applied to the obtained data. Patterns are sought in the data. Thus, this is referred to as an exploratory model of analysis.

The fundamental concept in this exploratory approach to data analysis is:

data = "smooth" + "rough"

The smooth is the underlying structure of a set of data values. It is the general shape of a distribution, that is, the regularity in the data. The residual data are called the rough. This is the deviation from the smooth. What distinguishes exploratory data analysis (EDA) from confirmatory data analysis is the willingness to examine the rough for additional regularity in the data. It is an openness that, instead of imposing a hypothesized model on the data, generates a model from the data.

In addition, reexpression of the data utilizing different scales of measurement may reveal unanticipated patterns. Therefore, one of the most useful approaches is to treat scales of measurement as arbitrary. Data can be reexpressed by any transformation so long as the discovered patterns can be related to the original data.

How EDA Works

After data values such as scores achieved on an examination are collected, they are ranked from lowest to highest, thus, forming a distribution. This distribution crosses a range of values within the data set. This distribution is best described in terms of three characteristics: location, spread, and shape.

Location refers to the point at which the distribution is anchored, usually measured by the median. The spread of the distribution refers to how wide the distribution is and how dispersed the values are. Shape refers to the type of distribution such as normal, symmetrical, skewed or bimodal.

Variables differ from one another in terms of all three characteristics. Identifying and describing each of these characteristics of the variables in the analysis is an important step in becoming familiar with data. To describe the data, numerous techniques are available, for example, numerical representation and summarizing the data. EDA makes extensive use of visual displays because 1) the shape of a distribution is as important as the location and the spread and 2) the choice of a summary statistic should be dependent upon the appropriateness of the statistics for the shape of the distribution.

Two easy ways to display data are to rank order it and to construct a histogram. A combination of these methods is the stem-and-leaf developed by John Tukey (1977). The stem-and-leaf makes two improvements over these techniques. First it is easier to construct and second, the stem-and-leaf illustrates the distances between points of the data.

To identify and describe in summary form the characteristics of a distribution, EDA utilizes 1) resistant measures of location and spread, 2) several summary statistics, rather than just one or two, and 3) visual summaries.

A resistant measure of location or spread is one that is relatively unaffected to changes, no matter how large, in a small proportion of the total data points. Statistics such as the mean and the standard deviation are nonresistant because they can be sensitive to a small number of values within a distribution.

The most useful resistant measures are the order statistics; these are based upon the rank order of the values in the distribution. The median is an order statistic and is utilized extensively throughout EDA. In addition to the median, lower and upper hinge points are used. The lower hinge is that point above which three-fourths the data values lie and the upper hinge is that point above which one-fourth of the data values lie. The distance between the lower and upper hinge is referred to as the midspread. Half of the data values are located in the midspread.

A technique to effectively represent data visually is the box-and-whisker plot, also developed by Tukey (1977). A box is constructed by drawing a vertical line forming the left side of the box located at the lower hinge; the right side of the box is located at the upper hinge, and a vertical line within the box represents the location of the median. Thus, half of all the data values in the distribution lie within the box, with one-fourth of the values between the median and one side of the box and another one-fourth between the median and the other side of the box.

In order to construct the whiskers, "X's" are placed outside the box one midspread from each of the hinges and are connected to the box with dashed lines, appearing as whiskers. Any data beyond these X's are marked with circles. Below in Figure 1 is an example of a typical box-and-whisker plot.

FIGURE 1
BOX-AND-WHISKER PLOT

Because almost 2.5 percent of all the data points in a normal distribution lie below the lower hinge minus one midspread and another 2.5 percent lie above the upper hinge plus one midspread, 95 percent of the values will lie within the range defined by the endpoints of the whisker. Thus, a distribution with more than 5 percent of the data points outside this range begins to depart from normality and can be quickly identified.

Following are examples of grade assignment using linear stanine equivalents of test scores and box-and-whisker resistant grade assignment.

Example 1 is 21 "spelling test scores":

96
93
91
88
87
84
82
80
79
77
76
75
74
73
72
71
69
67
66
65
61

Linear stanine equivalents are determined by adding or subtracting increments of .5 standard deviation (S.D.) to or from a base value of the mean and calculating floor values for each stanine.

Stanine	Calculation	No. of Grades	Letter Grade
9	M+1.75 S.D	1	A
8	M+1.25 S.D	2	B
7	M+0.75 S.D	2	B
6	M+0.25 S.D	2	C
5	M-0.25 S.D	4	C
4	M-0.75 S.D	5	C
3	M-1.25 S.D	3	D
2	M-1.75 S.D	1	D
1	--------	1	F

The letter grades are assigned at the discretion of the authors.

Example 2 uses a box-and-whisker to assign grades:

96------UPPER WHISKER
93
91
88
87
84------UPPER HINGE
82
80
79
77
76------MEDIAN
75
74
73
72
71------LOWER HINGE
69
67
66
65
61------LOWER WHISKER

In this example, B's are assigned to those scores beyond the upper hinge and inside or at the upper whisker; C's to scores at or inside the hinges; D's to scores below the lower hinge and at or inside the lower whisker; and F's to scores beyond the lower whisker. This grade assignment is similar to that done for linear stanines:

Letter Grade	With Stanines	With Box-and-Whisker
A	1	0
B	4	5
C	11	11
D	4	5
F	1	0

Example 3 changes the spelling test scores to skew the distribution downward. The three lowest scores are changed to "10's":

96
93
91
88
87
84
82
80
79
77
76
75
74
73
72
71
69
67
10
10
10

The linear stanines are re-calculated and letter grades are assigned to stanines in the same manner as they were in Example 1.

Stanine	Calculation	No. of Grades	Letter Grade
9	M+1.75 S.D	0	A
8	M+1.25 S.D	0	B
7	M+0.75 S.D	3	B
6	M+0.25 S.D	7	C
5	M-0.25 S.D	8	C
4	M-0.75 S.D	0	C
3	M-1.25 S.D	0	D
2	M-1.75 S.D	0	D
1	--------	3	F

Due to the lower mean and larger standard deviation of this data, the grade assignment is different from Example 1.

Letter Grade	In Example 1 (Stanines)	In Example 3 (Stanines)
A	1	0
B	4	3
C	11	15
D	4	0
F	1	3

Example 4 applies Box-and-Whisker grade assignment to the skewed data of Example 3.

96------UPPER WHISKER
93
91
88
87
84------UPPER HINGE
82
80
79
77
76------MEDIAN
75
74
73
72
71------LOWER HINGE
69
67------LOWER WHISKER
10
10
10

Grades are assigned as in Example 2. A comparison of the letter grade assignments of the four examples:

Grade	Stanine Ex 1	Stanine Ex 3	Box-and-Whisker Ex 2	Box-and-Whisker Ex 4
A	1	0	0	0
B	4	3	5	5
C	11	15	11	11
D	4	0	5	2
F	1	3	0	3

It can be seen that the Box-and-Whisker is much more resistant to the effects of wide outliers on the results than is the linear stanine method.

CONCLUSIONS

In summary, exploratory data analysis is an alternative method to evaluate a student's performance in the classroom. The major benefits in utilizing EDA are 1) the distribution better represents the typical student performance, 2) order statistics are more resistant to outliers, 3) the grade assignment process is more objective, and 4) the technique is intuitively understood by instructors and students alike.

REFERENCES

EBEL, R.L. (1972) Essentials of Educational Measurement. Englewood, New Jersey: Prentice Hall, Inc.

HARTWIG, FREDERICK. (1979) Exploratory Data Analysis. London, England: Sage Publication.

TUKEY, J.W. (1977) Exploratory Data Analysis. Reading, MA: Addison-Wesley.

VELLEMAN, P.F. (1981) Applications, Basics, and Computing of Exploratory Data Analysis. Boston, MA: Duxbury Press.

MICRO-ASSISTED SCHOOL MANAGEMENT
A TRAINING STRATEGY THAT WORKS

Charles A. Speiker
Susan L. Manuel

ABSTRACT

This presentation attempted to provide a detailed model for the training of school administrators in the use of microcomputers for management purposes. Several myths associated with staff development and training were dispelled, while a complete system of training was explicated. The training system was actually piloted with 60 administrators from 14 districts. Conditions and principles to guide the development of training programs were explained, as were the processes used during actual training sessions.

INTRODUCTION

Sages through the ages have claimed the common sense truth that is so easily hidden from the aimless thinker or the busier-than-watchful applier. Thoughtful reflection upon our education, training, and practitioner-soaked experiences also exclaim a truth . . . IF IT WORKS . . . DO NOT FIX IT and IF IT DOES NOT WORK . . . GET TO FIXING . . . BUT CAUTIOUSLY. Yet, school administrators will continue to drag themselves (and what is worse, their associates) to conferences, workshops, labs, quick-fix-slam-bam breakfast meetings with rubbery eggs and cold-grease coagulated sausage only to get answers to questions that are not asked and solutions in search of problems. Worse yet, if apparent eternal verity does surface, it soon disperses back at the office hovel, as water hitting a hot skillet.

Somehow the myth of progress (in the face of overwhelming evidence to the contrary) is perpetuated. People keep going to those sessions; speakers keep on talking; money continues to be used up; and, glowing reports eek their way out of the pen of the travel-weary writer. But, not too much is really being accomplished/implemented and secured over a period of greater than days or weeks after the revival session. (Aside: Because of the questionable quality of the quick-fix answers, maybe it is okay that we soon forgot what was espoused during the last hurrah!)

There was a need for more precision in training; a need for guides/conditions that would begin to assure the chance of back-home success for participants.

Guidelines for Training

These guides were meant to be taken in the context of a training session, not a workshop designed to heighten awareness. The guides did not speak to a conference that was grounded in an open-forum, give-and-take setting. These guides were training guides. They were directed to purposive and intensive outcome-oriented performance. They were used to shape any subsequent planning for training. And, some guides were seen as assumptions to be tested.

GUIDELINE ONE: Sound theory increases chances of success.

GUIDELINE TWO: Successful workshops produce successful back-home experiences.

GUIDELINE THREE: There is a difference between training and education.

GUIDELINE FOUR: Learning by doing has a more lasting effect upon people than learning by listening.

GUIDELINE FIVE: Concepts are formed through repeated trials and exposures to several incidences of an idea.

GUIDELINE SIX: Asking questions, rather than telling, will command attention and influence retention.

GUIDELINE SEVEN: People (this includes administrators) have a need to be entertained while they learn (learning as fun).

GUIDELINE EIGHT: People (busy administrators) need to see immediate application (results) for their investment of time.

GUIDELINE NINE: Activities and materials, rather than activities alone, will promote back-home success.

GUIDELINE TEN: Follow-up activities that encourage exploration and sharing will promote learning and prudent implementation.

Conditions for the Participants

Condition 1: Participants had at least a low level of computer awareness. Trainers were not concerned about traditional computer literacy.

Condition 2: Participants had access to equipment at their local sites.

Condition 3: Participants had a sincere desire to be trained.

Conditions for the Physical Setting

The conditions were acted-out during a series of workshops for 60 building principals, business managers, and school superintendents from 14 school districts. The workshops were set up with the following conditions:

1. Each workshop had a maximum of 14 people in each session.
2. Each workshop had two sessions, with each session lasting 4 hours, for a total of 8 hours of training.
3. Each 4-hour session was separated by at least 24 hours.
4. Each participant had a partner for the 8 hours at a fully-equipped learning station that included a computer, drives, monitor, and printer.
5. Each pair had an opportunity to spend at least 7 of the 8 hours at the computer.
6. The leaders were guided by tightly structured lesson plans with little room for variation.

Equipment Used

The Apple II with 48K and two disk drives and a IBM-PC with 64K and two disk drives were on site. Several varieties of printers were used including the Paper Tiger, IBM, and NEC.

Programs Used During Training

The trainers assumed that word processing, electronic spreadsheets and inventory management programs were appropriate training materials for each of the sessions. Visicalc, Data-Pack, Applewriter, Easywriter II and Mail-Pack were the commercial names for the training materials. Additional programs were mentioned including DB Master, dBase II, Wordstar, and Negotia-Pack. The programs were chosen because of their ease of use . . . devoid of highly technical requirements by the user.

Four programs were demonstrated. The four programs were examples of four functions that could be employed by administrators to make their jobs easier. Two programs were demonstrated at each session.

Methodology During Training Sessions

During the actual sessions, the particular methodology employed was a simple four-step process . . . and each process was repeated for each of four programs.

Step One: Demonstration

The first program was demonstrated to the total group. The participants were guided through program commands with a series of questions (ala Socrates). The programs were not completely exposed. Each participant received a diskette containing the program and a written program guide (short-form documentation).

Step Two: Worksheet

Each pair was given a worksheet (or a series of worksheets) on the program under study. The worksheets contained simple, yet detailed, step-by-step directions for simulated use. Participants used the worksheet in conjunction with the program on a diskette that contained "test" data. The simulated exercise was a time for instructors to provide tutorial assistance to pairs. Participants received as much guidance as needed during the use of the worksheet.

Step Three: Brainstorm

Small groups were organized when pairs completed worksheets. The groups were directed to list as many uses of the program as possible. A recorder turned in the group results for compilation and later distribution.

Step Four: Application

Pairs returned to their computer with another worksheet. This time they had to use the program with an application, as if they were back home. The worksheet had little or no direction. Test data on the diskette was no longer of use. The pairs had to struggle with this "real" application. Pairs could ask for assistance. However, they were given only indirect assistance. Questions were answered by more questions. The leaders struggled for learner independence. The application stage of the methodology was typically the most difficult of steps for participants. It was also this stage that would begin to assure success back home -- the learners could execute a program on new data with little or no eventual assistance.

The methodology was strictly followed, even when an isolated participant would openly object to the brainstorm activity as a useless activity. (Evaluations later suggested that the brainstorm sessions were as integral a part to back-home success as each of the other stages.)

Following a break, the second program was developed . . . using the same steps.

BUT . . . Did it work?

Thirty days later, it was reported (in the first follow-up session) that most participants had tried to use at least two of the four programs. (One program was an expensive commercial program.) Most participants were actively using at least one program or had trained an associate to use a program in his or her office or building . . . IF EQUIPMENT WAS AVAILABLE.

Written and oral evaluations suggested that the most glaring weakness was lack of time. However, that type of evaluation could be seen as a strong endorsement for the methodology and back-home success, because each person said they could operate each program with little or no assistance (aided by documentation).

Continued trials will be attempted to determine better activities, materials, and leadership styles (questioning techniques and sequenced demonstrations).

Acquiring Programs that were Used

Several wholesalers and retailers stock Visicalc, Applewriter, and Easywriter, as well as most other programs mentioned. Data-Pack, Mail-Pack and Negotia-Pack are available through Management Software Systems (MASS), Box 254, Boystown, NE 68154. DBase II is an inventory management program that can be obtained from Ashton-Tate at Culver City, CA 90230.

Additional information, as well as a book containing a complete set of all training materials (agenda, handouts, scripts), can be obtained by writing the presenters at:

Dr. Charles A. Speiker
Millard Public Schools
1010 South 144th Street
Omaha, NE 68154
(402) 895-8216

Ms. Susan L. Manuel
Educational Service Unit #3
4224 South 133rd Street
Omaha, NE 68137
(402) 330-2880

PROBLEM SOLVING SKILLS AND STRATEGIES: A GENERIC MATRIX FOR ELEMENTARY COMPUTER ASSISTED INSTRUCTION

Donna Stanger
Director, CAPS Project
Rochester Public Schools
1122 - 8th Ave. S.E.
Rochester, Mn. 55901

Both state and national assessments of 9, 13 and 17 year old students indicate a need to improve problem solving skills, particularly as they apply to everyday problems. Although elementary educators generally agree that problem solving skills develop from simple to complex and are related to the developmental growth of children, few have a clear mental picture or readily usable list of the skills and strategies that contribute to successful problem solving.

Computer assisted instruction can be effectively used to provide more and better opportunities for students to practice the skills and strategies involved in problem solving, but first there is a need to identify, define, and organize those skills and strategies in a form that is both clear and practical. This matrix is offered as a beginning schema. It represents a pragmatic, rather than a comprehensive attempt to understand and use existing theory.

Watching a group of first and second grade students play the computer game Bagels (students guess a 2, 3, or 4 digit number, using clues given by the computer in response to each guess), stimulated a mental dialogue that led to the search for and eventual development of a problem solving skills matrix. The dialogue went something like this:

What a fantastic experience this is! Look at the high level of motivation, the extended attention span, the willingness to persist!

Yes, but what are they learning?

Problem solving skills, logical thinking, many good things.

Which problem solving skills? What do you mean by logical thinking? "Good things" hardly qualifies as an instructional objective.

But look at the growth! When this student began she was only using random guessing, now she is beginning to use a process of elimination.

So now, we have two steps of a learning sequence: (1) Trial and error and (2) process of elimination. But what came before the first, and what could come after the second? What game should the children who are not yet ready for "Bagels" be offered, and what about those who have mastered the game? What next?

There the dialogue ended and the search began.

THE NEED FOR A GENERIC AND PRAGMATIC APPROACH.

Literature related to the improvement of children's skill in solving problems is fragmented and frequently represents a perspective that is far too singular.

Among the offerings from curriculum and instruction, often channeled through subject area disciplines are:

- "word problems" - mathematics
- critical reasoning - language arts
- creativity - fine arts
- logical thinking and experimentation - science

Added to these are the various approaches provided by education for the "gifted," educational psychologists, educational theorists, child psychologists, creative behaviorists, and so on.

Compounding the obstacles posed by fragmentation of the literature are those of academic loftiness. In a 1978 conference on "Practical Aspects of Memory," Ulric Neisser described two routes that have been taken in the study of memory: "a high road hope to find basic mental mechanisms that can be demonstrated in well-controlled experiments, and a low road to understand the specific manifestations of memory in human experience."

Neisser's remarks presented a strong argument for the study of memory using subjects in natural settings (in school, at home, on the job), pointing out that although the psychological laboratory was the easiest setting in which to work, it was also the least interesting, (and possibly the least relevant.)

These observations would apply equally as well to problem solving research. Often, experiments described are so tightly controlled, the topics so narrowed, the environments so sterile, that their relevance to elementary classroom education as we know it, seems remote.

For the educator who looks at the child and asks what can be done to improve problem solving ability and confidence in approaching and solving problems, a structure that is both generic and pragmatic is needed.

Generically, the structure and component items must translate readily to any subject area.

Pragmatically, the skills and strategies selected for inclusion must be conceptually accessable to the classroom teacher, and effective in the everyday life of the child.

Problem Solving Skill Matrix

MEMORY

- Mnemonic Systems
- Visual Association
- Self Testing
- Whole to Parts
- Creating a Contest
- Personalization
- Regrouping
- Auditory Aids
- Number of Items to Remember
- Sequence

COGNITIVE SKILLS

- Higher Order Rules
- Rules
- Defined Concepts
- Concrete Concepts
- Discriminations

COGNITIVE CONTROL STRATEGIES

- Simultaneous Scanning
- Identifying Multiple Solutions
- Examining Your Assumptions
- Working Backward
- Using a Model
- Focus Gambling
- Conservative Focusing
- Selecting Appropriate Notation
- Estimating, Predicting, Projecting
- Scanning for Clues, Hints
- Restating the Problem
- Analyzing
- Making Organized Lists
- Looking for a Pattern or Sequence
- Brainstorming
- Openness to Insight, Flexibility
- Successive Scanning
- Retrieval Strategies
- Problem Finding
- Information Gathering

CREATIVITY

- Fluency
- Flexibility
- Originality
- Elaboration

Based largely on the work of Gagne, the matrix reflects the influences of Paget, Guilford, Meeker, Bruner, Bloom, and Torrance. (Space prohibits the inclusion of definitions and examples for each skill and strategy.)

• MEMORY

The first column, MEMORY, is based on two years of both informal and formal research (funded by Title IV-C) involving students and teachers from the Rochester Public School District. During the first year, individual student interviews and classroom discussions were used to identify strategies intentionally used by students (K-6) when faced with a memory task. Other strategies, suggested by the literature review were added to form a basic list. This list was revised, eliminating: (1) those strategies to which students were already exposed, through existing curriculum, (for example: say or write each item to be remembered several times), and (2) those strategies for which no practical use could be envisioned.

Classroom activities to introduce most of the memory strategies and computer software (Apple II), to afford specific practice in using the strategies, were developed and tested. *(At the time of this writing, post-tests have been scored, but data analysis is not yet complete. Statistically significant student gains are evident from raw data for several strategies, and subjective teacher evaluations of both classroom and computer materials are overwhelmingly positive.)*

• COGNITIVE SKILLS

The second column of the matrix addresses the aquisition of Cognitive Skills as delineated by Robert Gagne.[1]

• COGNITIVE CONTROL STRATEGIES

The third matrix column, incorporates strategies that could be considered executive control processes, including four identified by Bruner,[2] and many of those typically utilized in mathematics, social studies, and science.

• CREATIVITY

The final column focuses on four divergent thinking skills as defined by Torrance.[3]

• CONCLUSION

This matrix represents an attempt to organize and clearly display the skills and strategies involved in successful problem solving. It can be used to (1) examine the skills and strategies of individual students, (2) examine existing curriculum in order to insure that students are exposed to experiences that foster development of a variety of problem solving skills and strategies, (3) examine existing software and classroom materials, in order to determine which skill or strategy is utilized, and (4) provide a structure from which to plan for the development of curriculum and courseware.

The identification, organization, and prescribed practice of specified problem solving skills is not intended to insure a student's proficiency in each, but rather to broaden and enrich the experiential base from which a student builds his or her own problem solving skill repertory. The aim is not to present a fixed problem solving model, but rather to promote utilization of an individualized, systematic approach in which the student eclectically establishes a model that is appropriate to the problem, based on his or her own skill and strategy repertory.

---

[1] Gagne, Robert M., The Conditions of Learning. New York: Holt, Rinehart & Winston, 1977, p. 34. (Gagne uses the term "intellectual." "Cognitive" is used herein to provide consistency.)

[2] Bruner, J.S., Goodnow, J.J. & Austin, G.A. A Study of Thinking. New York: Wiley, 1956.

[3] Torrance, E. Paul, Guiding Creative Talent. Englewood Cliffs, N.J.: Prentice and Hall, Inc. 1962.

SPECIAL EDUCATION TEACHER USE OF AN APPLE-BASED PROGRAM FOR WRITING INDIVIDUALIZED EDUCATION PROGRAMS (IEPs)

Christopher G. Stapleton, PhD
Independent School District No. 535
615 SW 7th Street
Rochester, MN 55901

ABSTRACT

The development of an extensive program in Apple Basic has facilitated the regular use of Apple microcomputers by teachers for the writing of IEPs in the Rochester (Minnesota) Public Schools. The initial disinterest or technophobia of the teachers involved has given way to generally favorable response to use of the program. There has been a side-effect of increased interest and appreciation for the instructional applications of the microcomputer.

The Legal Requirements for IEPs

Public Law 94-142, the "Education for all Handicapped Children Act of 1975", which took effect in public schools with the fiscal year ending September 30, 1977, has greatly increased the paperwork associated with special education since that time. Various forms must now be employed to document due process procedures at all key decision points in the identification, planning, placement, and delivery of services to handicapped children. Perhaps no document has had a greater impact in assuring this right to a "free appropriate public education" than the "individualized education program" (IEP).

The IEP has become an ever-present "fact of life" for special education teachers since 1977. It is a document that has increased the effectiveness of special education and focuses on the educational needs of each student. In the process, it has demanded a great amount of teacher energy and time to draft, implement, review and revise the IEPs. This is required for every handicapped student for whom the teacher is responsible. For those personnel who provided supportive service to mainstreamed students, i.e., itinerant special teachers, speech, clinicians, etc., the number of IEPs which they must write and manage is usually considerably greater than in the case of the special class teacher, who may be responsible for only 6 to 8. But in either case, the design and management of the IEP is a burdensome task, complicated by the requirement of multiple personnel involvement in the IEP staffing process.

Clerical Work Involved

Added to these demands are the clerical demands for producing the written document in a clear, concise and timely manner. A few teachers in our district have had the advantage of secretarial or aide help with getting the IEP into a typewritten final form, but the majority have had to produce the final document themselves. This resulted in a variety of final formats. Some IEPs were entirely handwritten on the standard forms. In other cases, the required information may have been supplied in handprinted format. When a large sampling of these documents was reviewed as an initial step in our Apple-IEP project, it was evident that the typical IEP in I.S.D. 535 varied widely in terms of its readability, arrangement of content, ordering and completeness of the annual goals, etc.

Use of the Apple to Facilitate IEP Writing

In an effort to determine whether the microcomputer could be of assistance in this situation, I.S.D. 535 initiated the Apple-IEP project. Beginning in 1981-82, the Special Education Department made programmed diskettes available to special education teachers for writing their IEPs on the Apple and printing them on either the IDS 440 or Epson MX-80 printers. Equipment was specially installed for this purpose at several pilot locations, and teachers were given the option of trying the newer approach or continuing to generate their IEPs as they had in the past.

The software was developed within the district by utilizing the services of an independent programmer, Mr. Phillip Brooks. Mr. Brooks is a chemical engineer employed by Minnesota Mining and Manufacturing. His programming work on the IEP was done at his home as an avocation. The major ingredients of the program were written in Applesoft Basic for the 48K Apple and DOS.

The program was designed to be as relatively simple to use as possible, menu-driven, and requiring a minimal amount of training for its utilization. The basic orientation for most users amounts to two hours of in-service with individual help provided beyond that if necessary. The majority of teachers did not require individual assistance beyond the initial orientation.

Revisions in the IEP program were undertaken as teachers tried out the initial versions. Their suggestions were utilized as the program was refined and debugged. The number of teachers using the IEP diskettes had expanded to nearly fifty in this second year of the piloting effort, 1982-83. This sample group constitutes about 35% of the total special education professional staff in the I.S.D. No. 535.

While the programming has remained in a developmental phase with additional changes still being considered at this time, no coercive demands have been placed on teachers for its use. A number of the current users are teachers whose assignments are not at the selected pilot school sites. These teachers have simply initiated their own use of the IEP software. Their self-initiated use of this approach to IEP development provides it own unspoken testimonial. Positive attitudes about the project appear to have increased as teachers become more confident with the program and gain familiarity with the equipment involved.

Evaluation of Teacher Reaction

Both initially and as more teachers utilize the microcomputer for writing IEPs, one of our prime concerns has been their attitudes about this approach to the task. They have been involved in writing IEPs since October 1, 1977, when the IEP was first required on every handicapped student who was included in our Child Count for Federal funds under P.L. 94-142. Subsequent to that time, teachers have become increasingly accustomed to producing these documents and more adept with the process. In spite of the increasing expertise, the IEP has remained a detailed, time-consuming and somewhat onerous clerical responsibility for special education teachers in our district.

Because of this, the Apple-IEP project was implemented in an effort to reduce the budensome aspects of the task and improve efficiency in the process. Since its beginning, it has been directed toward the teachers' point of view about what would be helpful. To this end, a three-page questionnaire was recently sent to all known users of the Apple-based program. This was designed to assess attitudes as well as identify suggestions for further improvements in the program.

Forty teachers responded to the questionnaire. While slightly more than one-half the users expressed some kind of complaint about the program or the equipment, a larger percentage (80+%) were either very positive about the program or in favor of it with reservation. Seventy percent of the respondents were in favor of expanding the piloting effort into other schools. These two percentages were considered the "bottom-line" in reflecting basic attitudes of users of the software.

The major complaints focused on lack of accessibility of the microcomputer and the general slowness of the microcomputer in processing large amounts of data input. It was the waits or breaks in the program due to processing that created some user annoyance. Part of the accessibility problem was tied to teachers' daily schedules. It was not so much that the Apples were not available, as it was a case where the teacher was not free to use them during the school day. As one teacher complained, "I cannot take the computer home, where I do a lot of the work involved in writing my IEPs."

Proposed Program Revisions

Among a number of minor program changes, quite a few users favored an option for accessing sequenced goals and related objectives. With computer storage of standardized sequences, teachers would only need to exercise a choice. The current version of the program provides users only the option of creating their own goal/objective sequences and then revising their lists by modifying, deleting, or adding. It is the appropriateness of the identified long-range goals and short-term objectives that is the key element of an effective IEP. It can be argued that the teacher-created goals and objectives render the more truly individualized IEP. Yet, these require considerable teacher thought and effort. Access to pre-existing goal banks through a computerized selection process holds appeal to a majority of the teacher respondents to the questionnaire.

Thus, we are presently considering an expansion of the existing program to include such an option. Pre-established goal/objective sequences would be stored on separate data diskettes accessed through the program. However, equipment limitations suggest that this may not be as time-efficient as teachers anticipate. Consequently, we wish to retain a dual option in this connection and continue to provide the user-written statements in this section of the IEP. Other program revisions that are actually being implemented are an earlier warning about the total amount of data loaded on the diskette in relation to its capacity, and an additional sub-menu for selective access to Section IX. These refinements are expected to further enchance teacher acceptance of the program for writing IEPs.

INCLUSION OF COMPUTER LITERACY IN DEVELOPMENTAL READING

Margaret F. Stier
St. Xavier High School
Cincinnati, Ohio 45224

Reading is a skill needed to survive in a media centered society. Computer literacy is also a skill needed to survive in an almost frightening escalation in technology. The future, as projected, is literally here; the use of the computer in all phases of society must be understood, not feared. Reading improvement classes progress at the optimum level of success if students see the relevance of skill materials. By this, I do not refer to the glut of teen-age books used in many courses, but to the need for a subject that will afford lasting improvement in any occupation or higher learning. Therefore, it is reasonable to assume that both efficient reading skills and computer literacy can be effected when combined.

St. Xavier high school, Cincinnati, Ohio, is a Jesuit college preparatory school. Developmental Reading is a required ½ credit course for all freshmen. The project described here was a direct result of personal experience with the confusion concerning computers that intimidates the general public. Research has shown that the best teachers are those who have experienced some frustration with learning themselves. After a few trips investigating the purchase of a computer and a short course in BASIC, I determined that the vocabulary, capabilities, and effect on society of the computer could not be ignored. In fact, what better place to teach computer literacy than in a skill area as advanced reading?

This unit could be taught in approximately three weeks, could satisfy student and parental interest in simple BASIC programing, could enable the student to follow the ever expanding advances in technology and moral implications as pertain to a sudden change in society itself. How will this effect students? One cannot hope for any control over technology without knowledge thereof. The overall goal of this unit would be to establish that the computer is a tool, a help to mankind; it must be used wisely, productively, with an awareness of the limitations of artificial intelligence for a balanced understanding.

The progression of project design developed rather quickly once it had been established that the administration would allow the experiment. The overall goal would be to prove that reading skills can be satisfactorily improved by using materials concerning the specific goals of computer literacy as exercises for speed and comprehension development.

SPECIFIC GOALS

1. To establish elementary understanding of computers per se.
2. To grasp vast potential of usage and phenomenal growth.
3. To create an awareness of certain moral problems causing a gap in "computer ethics".
4. To realize that there are limitations and inherent dangers associated with artificial intelligence.
5. To understand that computers will be used in any career; computer

careers are in demand, but all careers will and ARE utilizing computers.

In order to accomplish the above goals a sequence of activities was planned.

LEARNING OBJECTIVES

1. Student should understand what a computer is.
    A. How it works
    B. Different kinds
    C. Essential technical vocabulary needed to converse and evaluate
    D. Tests on vocabulary
2. Student should understand what elementary basic programing is.
    A. Flowcharting for problem solving (emphasis not on math)
    B. Experience execution of simple programs
    C. Construct a simple program by understanding rudiments of BASIC (PRINT, LOAD, RUN, REM, INPUT, READ, DATA, SAVE, GOTO, IF--THEN, GOSUB, TAB, FOR--NEXT, END)
    D. Execute one self-made program (hands-on experience)
3. Student will read WITHOUT ME YOU'RE NOTHING, THE ESSENTIAL GUIDE TO HOME COMPUTERS, by Frank Herbert.
    A. Exercise will dovetail as reading exercise/impart knowledge of what is happening across world--similar to TV revolution.
    B. Book is a practical explanation of situation designed to dispel fear of computer take-over.
    C. Test on contents
4. Student will have a written project in which research on current developments, moral, technological, or social, will indicate effects upon them.
    A. Articles/Xerox must substantiate findings.
    B. Written report must observe English department standards.
    C. Project necessitates using principles of efficient reading and application of knowledge gained.
    D. Grade given.

Personal research on computer literacy, attendance at the Cincinnati meeting of AEDS in May of 1982, and brainstorming with computer science teachers caused the evolution of the above outline. However, objective information attainment of computer knowledge would have to be measured as well as reading improvement. How to establish what questions should be regarded as vital was solved by a call to the University of Cincinnati. I asked what they recommended as a course in computer literacy at the undergraduate level. Upon finding out that Data Processing Capabilities and Facilities was a prerequisite for any study in computer science, except BASIC, I decided to sign up. After determining what university undergraduates were to achieve, I concluded that the freshmen at St. Xavier, gifted students, could learn a reduced but significant amount of the same material. Thus questions for the pre and post test were written.

A highlight of the computer literacy unit was a day-long exposition of seven different computers and capabilities by the St. Xavier Computer Club. One of the most pleasing and dynamic aspects of this service to fellow students was the "common sense" approach to the subject of buying a computer expressed in simple English. "Decide what you want the computer to do for you; there is no "best" computer.

Subjective evaluation of the project yielded the following conclusions.

1. Students were interested to the point of excitement concerning "learning".
2. Students helped each other; programming is a skill that comes harder to some than others. It brings out a cooperation that is very beneficial to personality development.
3. Students recognized that their math book held many clues to the program. i.e. simple programming is an option that is given by the authors. Many students used information in that book in our program--thus a bridge to another discipline was established.
4. The projects were excellent; they did not stop with articles from TIME and NEWSWEEK they interviewed people working with computers in many

fields. They read the journals of professions to find out how computers were being used, also quizzed parents who were in computer assisted or related fields.

Now to evaluate the post tests. As background, St. Xavier has had developmental reading as a requirement for eight years. The Nelson Denny reading test is used to evaluate comprehension and vocabulary progress, teacher-made tests for skimming and complex speed improvement record changes on materials they normally encounter in their academic situation.

The following scores are representative of the past seven years. Unusual advances by individual students are naturally compromised because these are averages of between 30-35 in a class.

KEY: Raw composite, grade level, and percentile will be noted as "RAW", "GR","* and words-per-minute will be denoted by wpm.

Nelson Denny		
raw	gr	*
pretest	15+	99
post-test		
111	15+	99
pre test speed		
skimming	complex	
334 wpm 67%	228 wpm 68%	
post-test		
605 wpm 91%	314 wpm 80%	

Nelson Denny		
raw	gr	*
pretest		
66	13	87
post-test		
76	13.6	93
pre test speed		
skimming	complex	
278 wpm	62%	212 wpm 33%
post-test		
535 wpm	86%	319 wpm 75%

Nelson Denny		
raw	gr	*
pretest		
58	12.6	81
post-test		
69	14.1	94
pretest speed		
skimming	complex	
270 wpm 64%	214 wpm 40%	
post-test		
527 wpm 88%	340 wpm 70%	

The following are the scores of the classes who experienced the inclusion of computer literacy this year.

Nelson Denny		
raw	gr	*
pretest		
70	13.3	90
post-test		
87	14.3	98
pretest speed		
skimming	complex	
343 wpm 68%	177 wpm 74%	
post-test		
471 wpm 91%	283 wpm 91%	

Nelson Denny		
raw	gr	*
pretest		
73	13.5	93
post-test		
87	14.3	98
pretest speed		
skimming	complex	
297 wpm 65%	190 wpm 62%	
post-test		
505 wpm 87%	316 wpm 82%	

Nelson Denny		
raw	gr	*
pretest		
53	11.1	71
post-test		
74	13.6	93
pretest speed		
skimming	complex	
259 wpm 54%	157 wpm 56%	
post-test		
531 wpm 84%	259 wpm 86%	

Nelson Denny		
raw	gr	*
pretest		
59	12.1	80
post-test		
74	13.6	93
pretest speed		
skimming	complex	
239 wpm 62%	152 wpm 56%	

post-test
520 wpm 88%   286 wpm 88%

Added to these statistics are the following preview skimming w.p.m. and percents. These have no pretests as students do not really know what preview skimming is until taught. In respective order:

612 w.p.m. 87%
733 w.p.m. 77%
608 w.p.m. 90%
1010 w.p.m. 80%

The results of the pretest on computer literacy were as expected; only 6 boys out of 134 passed, they all owned computers. However, having a computer in the house did not seem to enlighten 19 others. The post-test showed that 99 had assimilated enough knowledge to pass; 25 failed the test, but either doubled the original score or were low achievers in all other areas of the course. It is also important to remember that the content of the pre and post tests were based on what a university considered computer literacy, perhaps a bit ambitious for freshmen in high school.

There was a definite correlation between reading improvement and progress on computer literacy. Moreover, judging from records showing consistent improvement on general interest reading exercises in the past, the computer literacy unit did not deter normal progress. The question is not can computer literacy materials be used for reading skill development, but can more research and refining of handouts prove that such materials can actually cause the rate of improvement to accelerate.

Attitude is very important to reading skills. Finding subject matter that has carry-over into content area work is extremely important. Students like to learn about computers, logical problem solving is certainly employed in programing; does it not follow that this combination enhances the skills known as skimming (rejection of non-essential data), scanning (quickly locating literal answers) and complex reading (analyzation of factual material)?

The second semester classes will have finished this program by the end of April. More computers will be used for the hands-on experience, more skimming materials and programming exercises will be used. We believe we have found the most logical place to ensure that all students in St. Xavier high school will not only read well, but will be able to read and converse in "computerese"!

IMPROVING COMPUTER BASED LEARNING COURSEWARE
THROUGH APPLICATION OF COGNITIVE SCIENCE
LEARNING THEORY

Michael Szabo, Ph.D.
University of Alberta
Edmonton, Alberta T6G 2H1

Introduction

During the next decade, the production of computer based learning materials (computer assisted learning or CAL and computer managed learning or CML) will become a major activity among educational institutions and publishers. It has the earmarks of becoming a major business enterprise, heavily funded by private enterprise and governments and adopted by schools whose labor intensive costs have risen dramatically. The magnitude of the change is difficult to forecast because it will depend on society's ability to support education and that is largely a function of the health of the world economy. Regardless of the magnitude, this change will be far reaching and demands careful study. In this paper, I argue that we must learn to be highly systematic in developing CAL and CML systems. I will also take the analytic view which suggests that as a first step in the systematic process of producing computer learning materials, we examine carefully the learning processes of the students who will learn using these materials. Specifically, what "models" of the human learner do we hold and what do these models imply to developers of CBL systems?

Current Learning Theory

Teachers rarely consider models of the human learner since they usually deal with their students directly; hence there seems to be little need for models. Now however the computer gives the educator, through CBL, power to teach anywhere in the world. Many educators will develop teaching materials and never see the students who use them.

A second need for a model of the learner is that the computer makes it possible for educators to step up to new classes of educational problems. We have known for quite some time that learning rates among normal students varies in a 5:1 ratio. Now that the self-pacing of the computer makes it possible to accommodate this difference, we can incorporate it into our model. The generic problem of providing instruction which is sensitive and responsive to needs of individual learners has a potential solution in CBL.

Before examining specific components of current models of the human learner, I would note the three major learning theories of Western education. First is the stimulus response operant conditioning theory which has been "popularized" by B.F. Skinner. In its simplest form, it argues that learning is best when correct responses are reinforced and incorrect responses are extinguished. Operant conditioning largely treats the mind and its internal learning processes as a "black box." If one were to examine CBL materials today, there would be much evidence the developers were at least vaguely aware of reinforcement learning.

The second theory, popularized by Piaget, holds a developmental or maturational view. It posits several fixed and invariant cognitive levels through which learners progress. If the concepts to be learned are consistent with particular level of a given learner, the concepts can be learned. If not, meaningful learning cannot occur. Thus, using CBL to teach length concepts to a child who has not yet acquired "conservation of length" may be futile.

Although there are almost no CBL lessons which use maturation theory as a basis, there is one exception which is now world famous: the LOGO project of Seymour Papert. There is some indication that its use has been subverted by educators who use an operant conditioning rather than maturational framework.

The third major theory of learning has its origins in psychology, linguistics and computer science. It is called Cognitive Science and focuses exclusively on how the the mind functions during learning. Although cognitive science is relatively young, it has made substantial strides helping us understand some of the mysteries of how humans learn. It rests heavily on the model of the mind as an information processor. Cognitive science is rapidly becoming a more potent force in our understanding of learning as the reinforcement theories of operant conditioning fail to explain new phenomena (Kulhavy, 1977) and as Piagetian thought becomes integrated into it (DiVesta, 1982).

Current Models of the Human Learner and Origins

There are numerous learning systems, each having its own implicit and explicit model of the learner lurking somewhere. Rather than focus on any particular learning system, permit me to generalize from common experience and address the question - Given current educational practice, what model of the learner seems highly in evidence today? I have identified six 'inferences' of the human learner.

1. Reception Learning: Students learn primarily through receiving information. Almost all forms of instruction today stress the one way flow of information (information transmission) from the expert or authoritative content source to the student. Even the media or educational technology most commonly in use today to supplement lectures are themselves one-way oriented.
2. Copying Information: Students copy information which they receive. To the extent they have successfully "made a mental copy," we infer they have learned. Much evaluation of learning outcomes is based on how much the student remembers and significant amounts of instruction are "copying" oriented.
3. Equal Rates of Learning: By grouping students together in units called classes or sections and by presenting information to be learned at some hypothetical rate for the average student, we infer equal learning rates for all. So erroneous is this idea (as borne out by research and common sense) someone proposed to re-define intelligence as the rate at which people acquire information and knowledge.
4. The Learner as Competitor: Learners are placed in an environment where they compete for grades (and compete to learn?) The competitive aspect is further defined by awarding grades which are based on relative performance rather than preset standards. Students may "pass" whether or not they have acquired the necessary set of skills and knowledge.
5. Learning from Print: That learner best acquires information from the printed word can be inferred from the predominance of text in the classroom. This continues in spite of the inability of many students to read and the more elusive and persuasive inability of large numbers of our society to acquire deep conceptual understanding from textual symbolism.
6. Learning from Behavior Modification: Educators commonly state that learning must be "reinforced." This conjures up the vision of students eagerly awaiting confirmation of correctness and approval from their teachers. It also conjures up images of students who become increasingly dependent on educators to know if they are successful: a far cry from the goal of making students more and more independent.

Current educational sytems operate, perhaps unconsciously, using a model of the human learning which argues that students learn best when they receive information and copy information at the same rates, learn competitively from print materials, and are reinforced for correct behaviors. Observing typical instruction today would support the existence of that model.

Where did these educational practices come from? Can it be that educators responded to the industrial age which imposed a new order, one of regimentation, processing of raw materials, preparation for a lifetime at a specific factory job with efficiency quotas added to boot? Some historians answer yes. This system is clearly advantageous for adapting students to factory regimens.

Now that we are rapidly leaving the industrial age for the information age, is our model still appropriate? I think not and propose a new model, not based on speculation so much as knowledge we have presently gathered on what factors enhance learning. These factors to be listed below have two characteristics. First they are based on fairly solid research evidence and second, they can be manipulated by the teacher rather easily.

Proposed Model of the Learner for the Information Age

This model draws heavily upon the influence of cognitive science (of development) but does not shun behaviorism or maturation as a major factor in determining what is learned and how it is learned. The total model is presented as a series of propositions or learning requirements.

1. Transformation: Human learning is too complex to be described as making a copy of the world. Learning involves transformation; "...a transformation constructed from whatever cognitive, emotional, or motor interactions the person has had..." (DiVesta, 1982, p. 286). The result of many transformations is that a person's knowledge store and knowledge structure (ways of knowing) are continuously changing. It is no wonder that Cognitive Science literature makes extensive use of the analogy of the "information processor." In a sense, the remaining propositions are elaborations upon this information-processing theme.
2. Prerequisite Knowledge: Of all the variables that affect what a student learns, one of the most important is the amount of prerequisite knowledge and skills that student possesses at the beginning of a learning program (Bloom, 1976). Time and time again we have seen students fail because they do not have the prerequisites that are required for learning.
3. Learner Control: It is well known that students, particularly adults demand and take control over what they learn. There is some research evidence to indicate exercising control over the actual learning process results in better learning and longer retention. For example, practicing the skills to be learned helps the learner to master the concepts or skills being taught. If a student can increase the amount and frequency of such practice, he/she will learn more.
4. Competency Based: Students prefer to learn under conditions which will help them achieve competencies. In addition, they prefer to be compared with others in the class. In a truly individualized learning program, there may not be any 'others' at any point in time for comparison. A program based on competencies sometimes embodies a cooperative environment.
5. Mastery: The concept of mastery learning is simple and powerful and based on several research programs. In the applied sense, mastery means that the learner is expected to demonstrate knowledge or competence in any part of a course before being permitted to attempt the next part (Bloom, 1976). Mastery is very important for those courses in which successful learning of any part is dependent on successful learning of previous parts of the course.
6. Guidance: I am going to focus on guidance during the process of learning as opposed to career or other forms of guidance. When a student is engaged in acquiring competencies, there is a need to know which competencies have been met and which have not yet been met. Traditionally we rely on the student to make this determination: we use tests primarily to evaluate the student

and give a grade, rather than a diagnostic tool. There is evidence that that when a student is allowed to terminate instruction during a learning sequence, the student usually terminates before he/she has acquired mastery (Tennyson, 1982). If the information processing model is correct, the student will utilize this information and recognize the need for corrective action.

7. Learning Rates: When students are forced to complete a program of study at a pace set externally by the instructor, the tremendous variation in learning rates are not evident. It is only when we allow students to pace themselves that we see five-to-one learning rate variations (Bloom, 1974). The effect is so powerful that someone has defined aptitude as the time it takes a student to master something we have to teach!

   Due to variation in learning rates, when students are allowed to set their own instructional pace thaey complete the instruction in 20 to 50% less time, on the average (Kulik, Kulik, and Cohen, 1980). This finding is consistent and accompanied by no measurable loss in learning. If we are interested in increasing the learning efficiency of what we are doing, self pacing has much to offer.

8. Feedback: The area most profoundly affected by reinforcement ideas is feedback. It is most commonly believed that positive reinforcement after practice examples and tests will lead to better performance later on. The fact is however that feedback has its greatest influence when given after correct responses or practice. Feedback given after correct responses has little if any effect upon how much a student will retain.

   It may be that feedback works on using a transformational process in that it provides corrective information which the student can use to take remedial action. Further evidence comes from delayed feedback studies. When feedback is not given immediately after practice, students spend more time studying the feedback and score higher on retention tests, regardless of the 'reinforcement' they receive (Kulhavy, 1977).

   Does this mean we should not be supportive of our students when they have learning difficulties? Certainly not! The point is that we can be supportive and in addition use feedback to help students exercise some independence from teachers and learn more in the process!

9. Flexible Study Time: Probably the greatest barrier to further adult education is the unavailability of instructors at times when the learners can engage in instruction. Due to work and family schedules, adults cannot easily meet schedules which are convenient for instructors. Evening courses have proliferated in an attempt to ease this problem. The ultimate solution will come when students can complete the majority of their learning individually and on their own.

10. Questions and Practice Items: One way to utilize feedback is to provide for extensive practice questions and items during instruction. These 'embedded' questions have instructional value of their own however. It has been shown that when questions are embedded in text materials, and students process those questions, learning is significantly improved (Rothkopf, 1974). Also, questions given before the study of text materials can be useful in helping students focus on the important parts to be learned and de-emphasize the less important parts.

11. Instructional Objectives: Instructional objectives have been greatly misunderstood and mis-used by educators since their appearance in recent years. First, they should be treated as evidence that learning has occurred. Secondly, it has been conclusively shown that students who receive and study objectives before studying lessons will learn more as measured by tests (Duchastel and Merrill, 1973). The simple fact of having students study the objectives is an effective learning aid.

12. Graphic Learning: My own research into graphics has shown the power of graphics to improve learning. Specifically, we began with textual materials designed to teach concepts about the human heart. We added graphics which depic-

ted those points in the text we wanted students to learn and found that the graphics version resulted in significantly better performance in visual and conceptual criterion tasks (Szabo, Dwyer, and DeMelo, 1981).

We then took the study one step farther and added the same graphics to one version of the test. There we found for visually oriented tasks (location of the parts of the heart on a diagram), students did better if they had the visual test.

13. Individual Variation: Variations from student to student in the above learning requirements are substantial. Given these requirements are strongly related to learning, designers and developers of lesson face a substantial challenge in addressing these individual differences.

Conclusions

The advent of lessons developed for computer based delivery raises the need to critically examine the model of the human learner we are using. Specifically, requirements of learning in the information age. Fortunately, there is much current knowledge about human learning that will serve as a sound basis for formulating and testing new models.

Computer Based Aids for Curricular Planning, Test Creation and Processing

Gregory P. Thomas
Peter Wolmut
Multnomah County Education Service District
220 S.E. 102nd
Portland, OR. 97216

ABSTRACT

The Multnomah County Education Service District (MCESD) is charged with providing support service to 12 school districts with an approximate enrollment of 88,000 students K-12.

The Measurement and Experimental Research (M&ER) department provides a number of computer-based support mechanisms. The first supports the storage and retrieval of course goals and associated test items. Currently this system stores approximately 27,000 goals and 4,600 test items.

The second system provides a means of processing a variety of measurement instruments including achievement tests, criterion referenced tests, checklist instruments and surveys. Significant features include: interactive test and response sheet description; remote scanning of response sheets, scoring and reporting; updating of student competency records; and creation and maintanance of historic student test records.

This presentation describes these components in detail and allows participants hands on experience in the use of the various systems.

Curricular Planning and Test Development

To support curricular planning activities, the MCESD supports a computer based system which is currently composed of both goal and test item collections. These collections are organized around the following 13 curricular areas:

- Art
- Biological and Physical Science
- Business Education
- Computer Education
- Health Education
- Home Economics
- Industrial Arts
- Language Arts
- Mathematics
- Music
- Physical Education
- Second Language
- Social Studies

The number of goals varies from collection to collection ranging from a high of 4475 to a low of 195. The item collections also vary in their depth and breath of coverage. The numbers of items currently linked to curricular areas ranges from 0 to 3495. The variation in numbers of goals is a reflection of the current curricular thinking about the various areas. The variation in item numbers reflects the level of activity focused on the assessment of certain curricular areas/concepts.

Goal Collections -

Over the course of the past decade, the MCESD has coordinated the Tri-County

Goal project. Master teachers and other educators from the three contiguous counties, Multnomah, Clackamas and Washington, were charged with the development of a comprehensive goal structure covering all curricular areas. The current collection of 27,000+ goals represents the combined efforts of several hundred professionals in Oregon. The hard copy versions of this collection encompasses approximately 4,500 printed pages organized into 13 curricular areas within 16 volumes. This entire collection is maintained by the M&ER department in both hard copy and machine retreivable forms.

Item Collections -

At about the same time the goal collections were being assembled, the Northwest Evaluation Association (NWEA) was actively engaged in developing item banks in Reading, Language Usage and Mathematics. The items which are a part of these banks were statistically examined using Rasch techniques.

In addition to this item bank development, the M&ER department has over the course of years supported a number of other item generating activities. These activities have served to provide a mechanism focused on client district testing needs. The items generated as a result of this set of activities are examined statistically as they are used in a testing framework. At that point, the items are Rasch analyzed and kept, modified and/or deleted.

All items which are included are associated with goal(s) contained in one or more of the collections. This "linking" of items to goals allows the user to look for items as a independent activity or as a logical extension of goal development/selection.

System Uses -

Educators within the area have utilized the goal and item collections for a variety of purposes. It is possible to use a set of retrieval specifications to focus on grade, curricular area, key word(s), etc. By making these retrieval specifications broad or narrow, it is possible to "look" at a collection to gain ideas for future educational developments and/or changes. For example, the curriculum as currently defined may focus on a subset of a total set of possible variables. Developing a retrieval statement based on locally developed goals, the collections can be used as a sounding board to test the depth of curriculum coverage. This frequently has suggested additional areas that could be incorporated within a curricular plan.

Once a district has tentatively defined a curricular scope and sequence, the item collections can be addressed to create potential assessment instruments. Items may be retrieved independently of the goal collections or linked to specific goals. In this way teachers and administrators can look at items as "models" of the kind of assessment devices they wish to create. The item collections are not restricted to multiple choice and other paper-and-pencil styles.

This part of the total system is in place as a resource - not as a prescription dictating specific programs of instruction and assessment. By using the goal and item collections it is therefore possible to quickly search outcomes and items in an entire curricular area(s). It is important to point out that the access to and use of the system reside in the hands of the independent users. The staff of the M&ER department are at this point in a position to provide technical support and assistance to these users.

Test Processing and Reporting

One of the primary roles the M&ER department plays with client districts is the provision of a testing service. Tests which are utilized for assessing student performance are determined by the individual school districts. The M&ER department is, therefore, constantly having to examine its computer capabilities in order to meet the scheduled testing demands. Currently district wide testing programs operate in both the Fall and Spring of the school year. The number of students processed is quite variable from season to season and year to year. This current school year (82-83), we anticipate processing between 30,000 and 40,000 students.

From a processing point of view, a major amount of re-programming was common place whenever a new or different testing program was encountered. Because of budget and staff limitations a means of quickly adapting to changing demands was needed. The current test processing system was the result of this need. While this testing software package has certain limitations and constraints, the M&ER department has found over the course of the last year and one-half, that most tests can be accomodated.

The basic system limitations are:

- 200 items per test
- 20 goals per test
- 30 items per goal
- 5 choices per item
  (10 for surveys)

Test/Form descriptions -

The starting point in using this system is focused on the data collection form and a description of the actual test. Both descriptions are provided by the user, interactively. Data can currently be supplied to the system via Scantron compatible answer sheets; magnetic tape from another computer system; NCS scanable documents and "on system" data files. Using the Scantron data entry mode, the user can enter data from remote locations via the dial-in capabilities all County users currently have. Other means of data entry usually require intervention on the part of M&ER staff. Scantron compatible forms are custom designed and printed within the MCESD print shop. Costs vary with the complexity of the form but usually approach $25.00 to the point of plate production.

Entering the test description is also an interactive process. The software prompts the user for the various bits of information needed. This includes: number of test items; scoring key; scoring criteria; number of sub-tests; etc. Entry of a rather complex test description might take an inexperienced user 20 minutes. Once this test description is entered it remains on file within the user's logon until the user changes or deletes the description.

Data Entry -

At some point a physical test has to be created. Currently this operation is not computer supported beyond the point of item selection. Modifications are currently underway within the goal/item system to allow a user to select items, format a test and print the physical test using the already edited copy within the computerized item banks.

Once the test is created, however, the user has the options of scanning sheets which are either NCS (in-house) or Scantron (remote entry) compatible. Data is also frequently sent to the M&ER department by other LEA's for processing on magnetic tape. The data entry process establishes data file(s) which are named and used from this point forward - scanning is a one time process.

Scoring/Editing of Tests -

Scoring is accomplished by "pointing" the scoring program at the appropriate data file. The test description already provided is consulted in order to accomplish this task. The original test description can be changed and the scoring re-done without limit. This allows items to be keyed out at the users discretion.

Multiple marks are often encountered on answer sheets. An interactive edit feature allows the user to edit item responses as well as a number of other fields. Currently the score on both the overall test and sub-tests can not be modified by the user. If modifications are made, however, the user is able to rescore the data file to reflect these modifications.

Test Reporting -

Three "standard" reports can be produced by the user with a hard copy terminal device. These reports require that the user specify the particular report desired and the data file to be reported. There is no limit to the number of copies which may be produced by the user.

Custom reports are avaliable on an "on demand" basis. The MCESD currently supports several generalized report writers which are accessable to the system user. Using one or more of these utility packages, it is usually easy to provide custom reports within short time frames. Several users have also taken the time and energy to learn how to use these report writers and are actively engaged in generating their own reports.

Student Records -

The MCESD provides the capability for client districts to utilize the computer facility for student record keeping. Approximately 40,000 student records are currently maintained. The testing systems are currently able to update the individual student competency records, based on test performance. It is also possible to maintain student scores from any or all tests the student takes during the course of their enrollement. This system supports movement of the student between county districts as well as withdrawl and re-entry of students. That is, once the student is made a part of the system the record is maintained until the student is matriculated.

Other Capabilities -

A number of sophisticated statistical analysis packages are available to the user. These include an interactive Rasch analysis package; SPSS; Report writers and Graphic reporting systems. The user has the capability of using any or all such packages to examine the data file(s) which have been created during the data entry steps. It is also possible to retrieve historic information and produce reports designed to respond to specific trend types of questions.

EFFECTIVE USE OF A MICRO-BASED FILE MANAGEMENT SYSTEM

Nolan Tomboulian
Carteret Technical College
3505 Arendell Street
Morehead City, NC 28557

ABSTRACT:

Data processing is an activity that has existed since the advent of time and is not the result of "computer technology." FMS, a File Management System, is an effective tool to be used with a microcomputer system. A summary of uses and demonstration will interest those in administration, instruction, and private sector.

Data processing is not a new activity, it's something we all do every day. Consider these three examples.

Example 1: The Thrifty Shopper

The purchase of goods is a daily function, but usually a person will try to spend as little as possible. A clever shopper will absorb various inputs such as price, quality, and quantity, and then make a decision (process). The result will be that the best item is purchased and taken home to be stored for later use.

Example 2: The Decision Driver

There are established rules that drivers must follow to insure safety on the roads. A driver receives input from different sources through the senses and makes decisions. When the proper decisions are made, fewer accidents result.

Example 3: The Kitchen Cook

To provide something edible, (output) all the ingredients must be collected (inputs), and then the recipe (instructions) must be followed. If the dish is a success, the recipe is saved (stored) for later use, otherwise it is discarded.

In summary, there are four data processing steps: INPUT/PROCESS/OUTPUT/STORAGE. Of these, the first (and most important) step is INPUT.

Corollary 1: If you don't have it in, you can't get it out. In other words, you can't get back what was never there.

A computer is a device designed to perform basic mathematical operations and can electronically select and make decisions about data based on user supplied variables.

Data can be collected from many sources such as transaction terminals (cash registers), surveys, evaluations, and test answer sheets. The data is then massaged or PROCESSED. This involves any analysis performed on the input, including sorting, calculating, and summarizing. The results are then formalized into reports or OUTPUT. Finally, the data must be STORED for later use. If data can not be easily retrieved, then time and effort is wasted in the original recording and subsequent retrieval. Consider the time spent looking for phone numbers or reports that have been filed away for safe keeping, only to be "lost" in the catacombs of brown folders.

When discussing Data Management, there is a common platform of terms:

1) Character: one letter or number in a piece of data (byte).
2) Field: a unit of data; ie, name, address, sex, grade, phone, race.
3) Record: a grouping of fields in the same position.
4) File: a collection of records.
5) Data Base: collection of files with

redundant fields removed.

Example: File 1 - Name, address, sex.
Example: File 2 - Name, pay, sex.

A true data base would store the fields and establish an interface for the user to retrieve the necessary data. Most diskette-based micro-systems do not have the storage capacity to justify the creation and subsequent overhead needed for a true data base file structure.

Data Management Systems for microcomputers are usually "file oriented" in nature. This means that files are unique and that processing is performed on actual records in a file with no sharing of records between them.

The advantage of an Electronic File Management System over a manual filing system is that once the data has been entered, the system has the ability to massage the elements and extract information without interaction. This one ability can save many hours when it comes to retrieving data. Further, the system won't get frustrated with this tedious task.

In Data Processing, like in any other discipline, there are very few decisions that are absolute (there are always variables to consider). Users often ask salesmen or analysts for advice on the best program to purchase. To such questions, there is a well phrased three-word answer: IT ALL DEPENDS. All users will not have the same needs, and for this reason, no File Management System, or any other program, should be ruled out until the needs are defined.

FEATURES TO LOOK FOR:

There are two important features that any File System should possess. First, the program must be clear in its instructions and forgiving to bad responses. Secondly, there must be a balance between the needs of the user and what the program can provide without being a burden to operate. This sounds like a common sense rule, but all too often, a program is purchased that is so complex and hard to learn that it is actually counter-productive.

Corollary 2: If a system is hard to use, then used it won't be. In other words, disks should collect data, not dust.

THE SYSTEM IN USE:

At first, the acceptance of "FMS" was slow. The program was introduced with an inhouse workshop with representatives from most departments present. However, little interest was generated. The program saw limited use until those in the departments where it was not used saw the results that were produced in the one department innovative enough to try it. (Seeing is believing.) Reports and informational lists were timely, correct, and done with little effort.

The word "computer" can produce anxiety in many workers that have not been exposed to Electronic Data Processing. Before a computer can be effectively used, the user must feel in control. The computer is a "new tool" that can be used as a typewriter, calculator, or even a whole filing system. It will take awareness (literacy) to enlighten and thus elevate the user from the old position of "slave of the system" to the rank of "system master." Once the taste of computing power has been experienced, it takes very little to wean the worker off of the archaic typewriters and bundle of manila folders.

Corollary 3: Computers don't bite, they're given the bytes to store.

When this transformation occurs, a DP department will find itself beseiged with requests for specialized programs from all areas. It is at this point that a File Management System can save hours of program development since the system will handle input and correction routines and free the programmer to concentrate on any special processing or output requirements.

USES OF THE SYSTEM:

The File Management System can be used for just about any purpose a user can conceive. This system was designed to accept and correct input, and extract it for later use. It provides a foundation for all files, thus creating a consistency between programs. Ten uses of the system are summarized to exemplify the flexibility of the File Management System.

Corollary 4: Show me it can be done, and I'll find a way to do it.

Use 1: Faculty Roster/Phone List

A problem occurs in any dynamic organization where changes in membership occur. Rosters are necessary to keep information moving. Since it takes time to produce new rosters, they are usually out of date. Sometimes there is a need for these lists to be arranged differently; for example, by department, name, sex, special skill, etc. With little effort, the system can produce the output in any order and sort on up to six fields.

Use 2: Part-time Faculty Roster

Faced with budget constraints and hundreds of applications for part-time teaching positions, it was a major task to search through the job applications file to find the necessary instructors. Within a few minutes, criteria can be entered and the system will retrieve those names that meet the defined conditions.

Use 3: Class Offering List

Every term there are certain classes that are always offered. As a rule, this remains unchanged, thus the data can be saved and used again. If any data must be changed; for example, instructor, day, hours, or room, the data is changed where necessary and the new schedule is printed.

Use 4: Library Control

The Learning Resources Center has cataloged its magazines and books into FMS, thus providing an easy to use and current bibliographic list by subject area.

Use 5: Bookstore Inventory System

Maintains an inventory file. A point of sales program or "end of day updating program" is still to be developed.

Use 6: Registration

The program is used to store student data during registration. Using another program, the student data is extracted and placed into the class roster file. (A hard disk is really needed for full operation of this program.)

Use 7: General Student Reporting

To comply with federal rules, the Student Affairs Office must classify students by race, sex, academic level, handicap, etc. The system will provide the sorted lists for the federal reports.

Use 8: Financial Aids Tracking

Maintains data on students receiving aid and produces reports sorted on selected data.

Use 9: Annual and Sick Leave Report

This program provides for monthly updating to add and reduce the amount of leave that an employee has earned and taken.

Use 10: Faculty Evaluation

There is no one set of questions that will fairly evaluate all types of instruction. For example, laboratory classes will require a different set of questions than those for lecture classes. A base file has been established with over 150 questions. An instructor can customize an evaluation with questions that pertain to that subject area. Automated scoring and diagnostic output have been developed.

Though this is a brief list, it should be clear that the File Management System can be used for many purposes where data must be collected, stored, and recalled for various uses.

FEATURES OF FMS:

The system consists of two main sections: the File Directory Section and the Record Maintenance Scetion.

File Directory allows the user to define the fields to be used for DATA FILES, SORT FILES, and PRINT FILES. The Data File has the original data. The Print File contains the name of a data file and the format in which the data is to be printed. The Sort File contains a data file name and fields to sort on. All file definitions can be viewed for quick reference.

A problem with many Data Management Systems is that they don't allow the user to modify a file format once the data has been entered. This system will append fields to original records in an existing file and/or increase the length of defined fields. Fields can not be repositioned in the file, but they can appear differently through data print routine.

The system also provides a "Reuse" option that will take the format of one data file and allow the fields to be used for another data file.

The TRANSPOSE option will convert the data in a Sort File into a new data file.

Record Maintenance:

This module contains the necessary options to manipulate the data in a data file. The program will accept default values for fields to reduce user keying. When data exceeds the field length, input jumps to the next field.

Data can also be INSERTED into the file

between existing records if desired.

The data in the file can be FIXED (changed) or EXAMINED (looked at). The CLEAR FIELDS option will erase (replace) fields in every record of a file without prompting.

Records can be DELETED with an "@" as the first character of the first field, and will not be used in processing.

Two records in a file can be switched using the TRANSPOSE option. The SCAN option selects data from a file by keyed information for value(s) to search for. Data can be scanned in three ways:

1) First letter matches.
2) Relative values within fields.
3) Absolute values in fields.

The SORT and PRINT routines allow the user to batch seven and ten files respectively for processing, thus freeing the user from the time consuming task of system monitoring.

SYSTEM CONFIGURATION

APPLE II with 48K
1 or 2 disk drives
printer (80-130 columns) optional
FMS PROGRAM DISK (& MASTER CREATE data disk)
FMS DATA DISK

CONCLUSION:

Data processing involves four parts: INPUT/PROCESS/OUTPUT/STORAGE, and it is something that is done every day. A computer is only a device that enables a user to perform these tasks with less effort.

Some programs are cumbersome and, thus, counter-productive. There is no BEST program for anything. The criteria for selection must depend on the application and knowledge of the user.

Corollary 2: If a system is hard to use, then used it won't be. In other words, disks should collect data, not dust. FMS, the File Management System, does not require programming knowledge, yet puts the user in control. The system can be applied to Library Control, Registration, Student Tracking, Inventory Control, and Faculty Evaluation.

With this dynamic system, the types of applications depend on the ingenuity of the user. Some people will have doubts about any new system, but "seeing is believing" or Corollary 4: Show me it can be done, and I'll find the way to do it.

For any information on the above, contact the institution or write:

506 Simpson Street
Carboro, NC 27510

THE COMPUTER AS A TUTOR

Judith L. Toop
AUBURN ADVENTIST ACADEMY
5000 Auburn Way South
Auburn, Wash. 98002

ABSTRACT

The microcomputer can be used as an assistant to the classroom teacher in individualization of instruction. Programs can be written or purchased to repeat or supplement instruction given in class.

Greatest usefulness may be in catching students up when they have missed school or missed the point of some day's lecture. Computer programs may provide a vehicle for review for tests or assistance of one student by another in mastery of concepts.

The microcomputer will not replace the human touch of the teacher, but can ease his/her load of work and increase effectiveness in a time when the demand for excellence has not lessened, both the support for aides and assistants has diminished severely.

THE COMPUTER AS A TUTOR

It is a trying time for many of the nation's educators. Larger classes, fewer aides, more expectations by parents and administrators - all these and more are making it more difficult to produce a product which is predictably excellent. The teacher 's being asked to do an impossible task if limited to conventional methods of instruction. Technology can come to the rescue if given a chance. The computer can be used judiciously to ease the burden while promoting excellence.

As an innovation, the computer has taken off in the classroom. Thousands are in circulation and in daily use. As teachers of computer literacy, as promoters of hand/eye coordination, as baby-sitters, and as serious instructors, they have established themselves as an interesting and useful addition to the educator's equipment inventory.

I wish, in this presentation, to focus on the use of the computer as a tutor. This is not to lessen its other roles or suggest that they are unimportant. They are all part of the total picture.

NEED FOR A TUTOR

Military personnel overseas, travelling performers, ordinary people with business abroad--all these and others have employed tutors. They work with the student daily to keep them up with their classmates in conventional school systems and instuct them privately in their subject matter. The tutor must assess the student's needs and abilities and fit the schooling to the scholar. The tutor must be friendly and caring and able to cope with good and bad days with sensitivity and firmness. He/she must be knowledgeable in many subjects and willing to work hard at keeping current in educational methods and materials. Such is the human tutor.

The electronic tutor loses much of the human element. There is little ability to adjust to differing interests and abilities, no sense of comradeship, perhaps. Some semblance of communion can be established by the use of friendly error messages and encouraging lines which keep the student going in the program. But otherwise, the computer is a poor substitue for a humane and human teacher.

LIMITATIONS OF THE COMPUTER TUTOR

Because of the problems, the computer should never be used as a substitute for the classroom teacher, as a total educator. Course material consists of facts and figures, but also of feelings and attitudes, and these are not easily communicated by screens and disks and CPU's.

The computer tutor should assist the teacher and the student in solving several educational problems.

PROBLEM #1: "I DIDN'T UNDERSTAND!"

Especially in science and math, information is sequential. If a point is missed, the remainder of the procedure may be lost also.

Using a computer program which covers the same material in the same step-wise fashion, the student can pick up at his own pace and catch the drift of the whole class presentation. By running the program more than once he/she can cement the ideas into a sequence and understanding and, perhaps, take some written notes to supplement further study.

PROBLEM #2: "THERE'S A TEST TOMORROW!"

Panic can set in if the student feels unprepared. Some teacher review would be useful, especially on a one-to-one basis. When that is not possible, due to scheduling, the computer can pick up and allow for some drill/practice of the procedures or material to be tested.

Computer programs need to closely approximate the testing material if this method is to pay off for the student. If he feels that the computer time was not profitable, he may feel betrayed by man and machine.

PROBLEM #3: "I MISSED CLASS YESTERDAY!"

With larger classes, the teacher is likely to have someone absent every single day of the year. If he/she waits until all are there to present a concept or start new work, that day will never come.

When the missing student returns, it is crucial that there be some immediate steps to catch up, especially in sequential material such as math or science. The teacher's free time for tutoring and the student's free time from other appointments may not coincide. Enter the computer!

The screen has all the time in the world, and can be used for ten minutes or several hours, depending on how much time the student has or needs. For catch-up, this time and worry saver is especially useful. The computer programs must approximate the work covered in class and may need to be tailored to the specific text used by referring to certain pages or charts in the text used.

PROBLEM # 4: "MY FRIEND NEEDS HOMEWORK HELP"

Students gain from helping each other. Having explained something to another, you understand it better yourself. Having assisted someone else, you forge a bond of friendship which is sincere. Having gained help from someone else, you are more likely to offer to assist yet another friend.

If the computer can be such a friendly aide, then it is accomplishing something fine. As two or more students puzzle through some problems and the computer assures them of their correct responses or directs them when they are wrong, a bond of acceptance of the subject being taught and the people and machines involved is strengthened. This can't hurt the process of gaining information and learning how to learn.

PROBLEM #5: "I KNOW ALL THAT ALREADY!"

The brighter, quicker, more motivated student is a joy or a trial in the classroom, depending on several factors. When a subject is encountered on which a student is already well versed, or if some students need more drill/practice or explanation by the teacher, the quicker student can be left to cope with his boredom while he waits for the others to catch up. Mischief or dangerous mind-wanderings can be the result. The teacher can come to resent the student who he/she feels helpless to motivate.

The computer can rescue the situation. With programs which are beyond what the beginning class is accomplishing, the advanced student can be challenged. Or, if he/she has a special interest, programs can be secured which will take hold of that interest and direct it into profitable expressions. The brighter student who is so inclined might even originate some computer programs of his own to assist the other students or the teacher in the class in which he is involved. The possibilities are many, though they take careful planning and considerable forethought.

IN CONCLUSION . . .

The computer has entered our lives to stay. It can be a tool to assist, enrich and otherwise make possible the tasks we are set to accomplish. If used properly, it can effectively substitute for non-existent aides and fill in for the extra time it would take a teacher to assist a student in special ways.

The teacher will never be replaced by the computer. The personal human touch will always be superior. In using the computer, the teacher must be aware of this, and supplement the computer's possible coldness by special attention during instruction times and in post-session wrap-ups as he/she touches base with the tutored student.

The computer is here to stay. The computer tutor can be the unpaid hero of the modern classroom.

MICROCOMPUTER APPLICATIONS FOR SPECIAL ED AT WAYNE COUNTY INTERMEDIATE SCHOOL DISTRICT, MICHIGAN

Greg Turner
Wayne County Intermediate School District
P.O. Box 807
Wayne, MI 48184

Wayne County Intermediate School District of Michigan (WCISD) has been engaged in a program of microcomputer applications for Special Education since 1975, when a cooperative research agreement was undertaken with Michigan State University (Artificial Language Laboratory) to "provide modern technological aids to communication...for students who experience communication handicaps." Since 1978, these efforts have been conducted entirely in-house in the WCISD Data Processing Department. Originally funded by U.S. Department of Education Title-I monies, the project is now supported out of the ISD general fund.

From 1975 to 1981, the program was concentrated on the design and fabrication of prototype devices for the students of the so-called Experimental Classroom. These students were seven cerebral-palsied teenagers with moderate to severe physical disability including non-speech, who were drawn from SMI (severly mentally impaired) and TMI (trainable mentally impaired) classrooms, on the basis of their comparatively - high-functioning receptive language abilities. Instructed in the use of the Blissymbolics alternative symbol system, the students were then "matched" in terms of their varying physical abilities to adaptive input devices (touchpanel, optical head pointer, morse code). These adaptive input devices then served as the front-end to the microcomputer-based language handling system, permitting the students to select or spell words, form sentences, store same for later retrieval, and output these to voice synthesizer, LED twenty-character display and twenty-character thermal printer. These devices were battery-powered and housed in stock wood laminate cases and were intended to be wheelchair mounted, but a variety of problems (weight, battery life) militated against this and the devices remained tabletop. In this context, the devices were used primarily in an instructional framework, where the students were able to work independently on worksheets and write tests (via the printer), and to participate verbally (via the synthesizer) in large group sessions, question and answer, etc. In 1981, when these students returned to their locals (with devices), five of seven were reclassified EMI (educable mentally impaired) and of these, three are presently in POHI (physically or otherwise health impaired) classrooms in general ed highschools.

Since June 1981, work has continued on a second-generation of communication aids. Currently, two separate systems have been approved by the WCISD Board of Education for resale to educational and rehabilitative organizations and to parents/guardians of communicatively handicapped students.

The Apple Communication/Education Station (ACES) consists of the Apple II, monitor, disk drive, printer, Votrax Type 'n' Talk speech synthesizer, custom 22 x 14 touchpanel, and custom serial-parallel i/o board. Menu-driven software permits non-computer professionals (eg. teaching staff, speech therapists, etc.) to create diskette-stored vocabularies, edit same, and associate these with user-defined selection cell grids imposed on the 22 x 14 touchpanel array. These vocabularies can then be represented (via overlays) on the touchpanel as desired, i.e. photographically, pictographically, alphabetically, etc. Besides the basic touch and talk program, staff can also run a tutorial program that randomly selects vocabulary items and queries the student-user for position. This system is intended primarily for use as a diagnostic and therapeutic device by classroom professionals.

The Hermes (named for the Greek god of Communication) is a portable, battery powered device with touchpanel input and synthetic voice, LCD display, and miniature impact-line printer outputs. Vocabularies are custom programmed. Housed in the custom fiberglass case are the WCISD designed electronics, consisting of the microcomputer board, output board, and touchpanel board. These boards were prototyped (breadboard and wirewrap) and laid out (line drawing) in-house; actual printed circuit board manufacture was farmed out to a local firm. As such, the boards incorporate several power-saving features (CMOS integrated circuits, power-down processor and EPROM), permitting considerable downsizing of batteries vis-a-vis the previous generation of devices. Hermes fabrication is also presently in-house and we are currently in the middle of a "mini" production run of twenty-five devices. These are being placed with a variety of Special Ed users: physically-handicapped, mentally-impaired, autistic, and hearing impaired.

For the future, we look forward to the development of additional devices. Currently, a micro-cassette drive is being interfaced to the Hermes to facilitate end-user creation of vocabularies and to support a cassette-based curriculum of expressive language acquisition. To these same ends, text to speech software is under development for the WCISD voice synthesizer.

These are extremely exciting times that we find ourselves in, when the so-called high technologies of modern electronics and micro-computing can be manipulated by educational organizations for the benefit of the special education student population. We at WCISD are extremely proud to make our contibution to the endeavor.

Computer Music Applications in Education: The Soundchaser Computer Music System

Anthony A. Turrietta
Portland Community College, Rock Creek Campus
12000 S.W. 49th Ave., Portland, Oregon 97219

Cheryl R. Turrietta
University of Portland, Performing & Fine Arts
5000 N. Willamette Blvd., Portland, Oregon 97203

Computer Music has crossed many disciplines throughout its history, including physics, engineering, psychology, audiology, and special education. An affordable, comprehensive system is now available that has diverse applications in educational environments. The Soundchaser Computer Music System includes the "Music Editor," "Player," "Instrument Definer," Performance Software ("Turbo-Traks") with polyphonic keyboard, "Music Tutor" and "Notewriter." Among its applications within the field are: live ensemble performance, composing and arranging, creating original instruments, music transcription, music fundamentals and theory drills, and a Computer Music course that has attracted students without experience in music or computers. Other applications include general computer literacy, a musical instrument for handicapped students, and in courses in the physical basis of music.

The "Soundchaser Computer Music System," manufactured by Passport Designs Inc., is a comprehensive package designed for the Apple II, II Plus, IIE or Franklin Ace 1000 microcomputer. The complete system includes a sixteen voice digital synthesizer (two IC boards), Mountain Computer's "Music-System" software, a forty-nine key polyphonic piano-type keyboard with interface, "Turbo-Traks" Performance software, the "Music Tutor," "Notewriter" and documentation; all of which can be incorporated into existing traditional music programs.

Turbo-Traks Performance Software

"Turbo-Traks" software is a sophisticated program consisting of four sections described below.

The Performance Section loads sixteen instrument definitions, called "Presets," into memory that can be played as is at the keyboard or modified in real-time by the user. Each preset definition contains waveforms (spectrum defined by user), amplitude envelope specified numerically as standard Attack-Decay-Sustain-Release (ADSR) for each oscillator, octave setting and volume control. Additionally, there are four Low-Frequency Oscillators (LFO's) that may be applied to create frequency modulation effects, such as vibrato, phasing, sirens, etc. The rate, amount and waveshape of the LFO can be determined by the user, as can all other Preset parameters. These Presets may be stored in groups of sixteen Presets called "Sound Files," or, musically speaking, original

'Orchestras.'

The Wavemaker Section is used to design a waveform for each oscillator and the LFO's. The Create command sets up a harmonic graph for specifying which harmonics are present at what amplitudes. Edit command allows waveforms to be drawn in directly using game paddles. Previously saved waveforms can be loaded into the Wavemaker and combined in new ways. The Move command allows for fast creation of complex waveforms by 'moving' a summed waveshape in place of each harmonic present. The keyboard can be re-tuned and the frequency cut-off of the Smoothing Filter can be controlled from the Wavemaker.

The Recording Sequencer Section allows up to sixteen tracks of music to be recorded, saved or played back. All Preset parameters can be altered live while playing back a 'Traks' file, including Tempo of playback.

Disk Section operations are to view Catalog, Load, Save, Rename, and Delete individual "Presets," "Sounds," "Wave," "Traks," and "Tuning" files.

Because of the System's flexibility, its applications in educational settings are numerous and diverse. It can be used in live ensemble performance much the same as an electric piano or harpsichord. Original instruments can be tailored for the situation by students and performed by keyboard players. It is also an economical instrument for live Theater and Dance productions, and Video-Tape or Film Sound Track composition.

As a composing instrument, sound experimentation and music scoring and arranging can be done quickly and systematically, unlike using conventional notation and rehearsals or analog electronic music systems. (q.v. references #4 and #9 for more information.)

The Music Tutor

The Music Tutor is an effective tool for building ear-training skills for musicians of all ages. It can be used in Music Theory I, Music Theory II, and Fundamentals of Music courses, or by any musician interested in developing necessary listening skills. The drills included in the most recent version are: Melody memorization, Interval and Chord Identification (including inversions), Pitch Matching and Chord Tuning.

The tutor package contains three keyboard overlays and three program diskettes: Intervals Disk, Chords/Tuning/Melodies Disk, and Teacher's Disk. No additional hardware is required to run the programs. High quality sound is achieved by hooking up to a standard home stereo system.

The drills are a series of self-paced, user-friendly programs that instruct the individual through "Free Practice" or "Guided Practice" exercises. Responses can be made at the Keyboard, by numbers corresponding to the answers, or by names of notes. Evaluation and results are shown to the student on a Testing Chart, which is helpful in monitoring the progress of the user.

Certain parameters of the drills may be altered using the Teacher's Disk; such as order of drills presented, percent correct required to test at the next level, number of intervals presented at each drill, and the tone produced.

Overall, the Tutor is well thought out and considerable in scope. It can be made challenging enough for students with perfect pitch or patient enough for novices in ear-training. (q.v. reference #7 for detailed review.)

The Notewriter

The Notewriter is a real-time, monophonic music transcriber. A single voice is played at the keyboard against a "click" track and immediately displayed at the screen. "Notelist" files can be saved, edited (such as adding repeat signs, inserting or deleting notes, etc.), printed out on a Dot Matrix with Grappler interface and SilenType printers, or played back at any time. The Notewriter Editor also allows for changing key, meter, transposing, or entering a new piece. The system is faster and the print-out superior in appearance than most hand written copy work. Its most suitable applications currently are for transcribing and transposing instrument parts, and allowing students who have not yet been trained to notate compositions to do so. The camera-ready appearance makes it particularly useful for insertion of musical examples into typed text.

Notewriter Version 1 is the first of a long developmental process planned by the manufacturer leading to a real-time, polyphonic music transcriber.

Mountain Computer's "MusicSystem"

The "MusicSystem" software, manufactured by Mountain Computer, Inc., is included with the Soundchaser system and provides alternate ways of generating music. The Music Editor is used for entering musical scores (up to sixteen parts) using simple code, menu selection, game paddles, or an attached light-pen. "COMP" files are then saved to be played by the computer in the Music Player. There are ten instruments supplied with the system that can be assigned to different parts in the Player, or original instruments can be created in the Instrument Definer section of the program. The Music Merger allows for shorter files to be merged together, creating a longer file that can be played by the Player without interruption. Graphic displays include waveforms, harmonic graphs, 'Amplitude Histories,' 'Frequency Histories,' and standard clef, staff, meter and key signatures, and musical notation. Parts can be printed out one at a time from the Editor on a SilenType printer.

MusicSystem software is most useful for introducing computer music to students of all ages and to students whose composition and creative capabilities exceed their keyboard technique. (q.v. reference #5 for review.)

The Apple

Other possible uses of the Soundchaser and the computer within a typical College Music Program are many; ranging from record keeping, planning and forecasting Festivals and word processing to including visual and aural representations of the Overtone series in a Music Appreciation Class.

One example currently in progress is a software program designed to cut the multiple staff hours required within a music program to plan, develop and execute Music Festivals. These festivals usually range in size from one to five hundred participants, and range in time from one to twenty hours. The program can handle time, location and sequence of events from fifteen minute modules to one hour modules. The organization includes sizes and types of rooms and equipment, numbers of students, directors and adjudicators; all of which must be shifted smoothly in exacting time increments throughout the day. The bulk of the work can be completed in one meeting; continual updates can be implemented immediately. (Changes can be made up to the last minute.) For every necessary alteration that is made, the schedule for the entire festival is effected. The Apple can accommodate these inevitabilities with ease.

Another example currently being developed within the College Band Directors National Association (CBDNA) is a software program to help develop intricate designs for Marching Band routines and drills.

Conclusion

In addition to the Soundchaser's use in traditional music courses and possible use in electronics and physics, a computer music class attracts students from diverse disciplines and backgrounds. Performing Arts students can become acquainted with computers through a familiar musical language, and, conversely, those with no background in music can be introduced to sound appreciation through the computer. Students who are physically handicapped can study the elements of sound, perform and compose music with the computer; as physical training and agility is not a prerequisite to creating music with the Soundchaser system.

Real-time audio feedback combined

with graphic displays can contribute greatly to learning concepts of the elements of sound and composition. When creating an original instrument, for example, each change in any parameter can be heard immediately, thereby improving critical listening, analysis and synthesis skills.

The Soundchaser Computer Music System is the most cost-effective, versitile instrument available today. It costs much less and has more extensive and diverse applications than do acoustic instruments and analog synthesizers. The programs are easy enough for very young students to use and sophisticated enough for advanced research and composition by professionals.

SELECTED BIBLIOGRAPHY

1. *Arenson, Michael A. & Fred T. Hofstetter, "A GUIDO System and the PLATO Project," Music Educator's Journal, January 1983, p.46.*
2. *Buxton, W. & W. Reeves, G. Federow, K.C. Smith, R. Baecker, "A Microcomputer Based Conducting System," Computer Music Journal, Vol. 4, Number 1, 1980, p. 8.*
3. *Chamberlin, Hal, Musical Applications of Microprocessors, Rochelle Park, New Jersey. Hayden Book Company, 1980.*
4. *Moog, Robert, "The Soundchaser Computer Music System," Byte Magazine, December 1982, p. 260.*
5. *Moore, Robin, "Mountain Computer's MusicSystem," Byte Magazine, July 1981, p. 60.*
6. *Tjepkema, Sandra L., A Bibliography of Computer Music: A Reference for Composers, Iowa City, Iowa University Press, 1981.*
7. *Turrietta, Anthony A. & Cheryl R. Turrietta, "Soundchaser Computer Music System: The Music Tutor," Apple Journal of Courseware Review, Issue 2, 1983.*
8. *Turrietta, Cheryl R., "Building a Music System," Music Educator's Journal, January 1983, p. 52.*
9. *Turrietta, Cheryl R., "Positive Notes on Computer Music," T.H.E. Journal: Technological Horizons in Education, January 1983, Volume 10, number 3., p. 74.*
10. *Upitis, Rena, "Milestones in Computer Music Instrction," Music Educator's Journal, January 1983, p. 40.*

COMPUTER LITERACY PROGRAMS
and FACULTY DEVELOPMENT

James S. Vinson
Dean, College of Arts & Sciences
and Professor of Physics
University of Hartford
West Hartford,CT 06117

ABSTRACT

The University received a National Science Foundation Grant under the CAUSE Program in the fall of 1981. The Grant was entitled, "The Improvement of the Computer Competency Level of Arts and Sciences, Basic Studies, and Engineering Students." This grant allows us to explore ways of improving computer competency and computer literacy on the campus for both students and faculty while also providing specific opportunities for course revision and faculty developments.

The project focuses upon the need to strengthen the computer literacy of a wide variety of students. Since computers have revolutionized the ways in which we work and lead our private lives, vote, communicate, and educate, it is essential that universities respond with curriculum revision. Academically, the computer has become a fundamental tool of the intellect both for the teacher and the student. In the sciences the computer has become an integral part of virtually every discipline. Computers control experiments, store data, analyze data, and manipulate large data bases. However, at present most university graduates have an insufficient knowledge of computers and of their widespread use throughout society.

Our long range goal should be to graduate students capable of understanding and initiating the use of the computer in an undirected environment. The first step toward meeting this goal must be the establishment of a minimum computer competency for the students. In this project the minimum competency for all students is defined as the ability to interact with the computer in an intelligent manner. Specifically:

1. The student should have the ability to formulate a problem in a specific discipline and in a form suitable for solution by a computer. As such, the use of the computer requires an individual to approach both the formulation and the solution of a given problem, rationally, systematically, and logically.

2. The student needs to develop the perspective that the computer can be

> a tool to be used directly in the laboratory to record, store and analyze data during an experiment, and therefore is a key element in experimental design and decision-making.

We have set a higher standard for engineering, natural science, and mathematics majors because these fields require greater computer sophistication. The insights to computer use developed through the preceding objectives must be reinforced and become more focused. The student must be able to:

> incorporate the computer into his learning environment through applications both intricate and complex in his special field of expertise. At this level, also, students will develop a recognition of the potential and limitations of the computer an an analytical tool.

Several related activities will help us meet these competencies for science, engineering, social science, and general liberal arts students. Of course it is also necessary to have adequate hardware facilities and computer terminal access to implement such a program. First, we have developed a training program that will instruct faculty in programming and computer interfacing and will prepare them to use the computer in their courses. Second, a number of course revisions have been completed at the introductory level and a new introductory course has been added. All introductory courses in the natural sciences demonstrate to students the use of the computer within an experimental problem solving context. Simulations are used to aid in the understanding of complex problems and systems. Finally, the new introductory course introduces all students to programming and the impact of computers upon society. Third, in order to ensure that science, mathematics, and engineering majors are competent in using the computer for complex operations, various upper level required courses have been revised. These revisions will emphasize data collection and analysis.

Perhaps the most important aspect of any such project is the development of faculty competency. The faculty training component of this project prepares the faculty for the use of the computer in specific courses and generates in cohesiveness with respect to the implementation of computer related education across the curriculum. Specifically, the faculty training program consists of a two week summer course on programming and basic computer concepts and a two week summer course on interfacing. Each day consists of a 1.5 hour lecture, four hours of supervised laboratory time during which the faculty work on specific programming problems using a microcomputer and minicomputer, and a final hour for summarizing and discussion. The programming course is open to all interested faculty while the interfacing course is available to the science faculty. This interfacing course is more advanced and covers the fundamentals of machine language programming during the first week and the actual interfacing to instruments in the second week. After participating in such courses, the faculty member is prepared to revise specific disciplinary offerings within the curriculum.

Evaluation is a continuous and ongoing aspect of this project and addresses several concerns. First, is the project design proving effective. Second, has there been a growth in faculty understanding, ability and willingness to use the computer in their courses. Finally have student competencies developed in the areas designated in the objectives.

AN INTEGRATED CLASSROOM LEARNING SYSTEM (ICLS) USING TECHNOLOGY

Gary D. Volding
Ideal Learning Inc.
1044 South Fuller Street
Shakopee, Minnesota 55379

The project in the Shakopee, Minnesota Junior High School is a joint venture between the Shakopee Public Schools and Ideal Learning Inc. of Shakopee, Minnesota.

Initially a number of educational needs were identified.

- Gifted programs
- Instructions for absent students
- Class sizes too large
- Small schools lacking upper level courses
- Accountability of student achievements
- Budget problems
- Shortage of teachers(math and science)

After identifying the needs the question of how to meet these needs was explored. The following conclusions were drawn.

A new delivery system for instruction would be designed. Technology would play a major role. The instructor would remain the focal point.

The area of technology was investigated to determine what was available and how it could be implemented. The final decision was to establish a micro computer lab on a network system. It was determined that the only way to have a multi-user lab was to network. In turn the approach to software development had to include floppy disk and hard disk capabilities. The role of video disk was explored but it was determined that the concept would be a part of long range planning but not a part of the initial plan. Secondly the role of the teacher to be full time instructor and prescriber of what each students needs would be.

The system is designed to minimize the teachers time spent on monitoring and scoring tests and daily assignments.

After the role of the teacher was determined and the technology was identified, development of software was began.

First an operating system was developed which will work with software developed in any discipline. The system is designed to include instructions, assignments testing capabilities, and a management system for assignments and testing. The system is designed so gifted students may move at their own pace in the lab setting without attending lecture sessions.

The system also has the capabilities of assisting students who are having difficulty in a particular subject area. In the Shakopee Junior High School algebra classes the students meet with the teacher one day for instruction and the next day they spend in the lab doing assignments and testing. They continue on an alternating day schedule throughout the year. This has allowed class size to be reduced and provides the teacher with more time for instruction.

The main feature of the system is the flexability in how a school system may use it.

Courses in eighth grade math, first year algebra, advanced algebra and trigonometry will be available for the 1983-84 school year.

Curriculums in seventh grade math, geometry, math analysis, calculus, chemistry, physics, and foreign languages will be written during the summer of 1983. This plan is designed to produce full year courses in all disciplines so that schools may fully utilize technology in the quest for improving the education of the youth of America.

TORT LIABILITY FOR INSTRUCTIONAL TECHNOLOGY

A Standard of Care for the Use of Computer-Assisted Instruction (CAI) Within the Instructional Environment of the Classroom

Brent E. Wholeben, Ph.D., E.M.T., Associate Professor
Department of Educational Administration and Supervision
The University of Texas at El Paso (El Paso, Texas 79968)

NEED FOR CAI LITERACY TEACHER TRAINING

The use of some form of computer-assisted instruction has increased markedly within the classroom environment of the school during the past few years. All predictions are that this trend is not another passing fancy of the educational establishment. In fact, many instances show the desire and background of the teacher supporting introduction of microcomputerized instructional technology within classroom activities of the child, as opposed to the established policy of the building principal, district superintendent, or school board members. But whether the impetus for the use of computer-assisted instructional methods arises from teacher or manager, the need for adequate training in order to insure valid and reliable utilization of this technology must be recognized.

While the Constitution of the United States recognizes the right of every child to an education, individual state laws have established this global right as mandatory requirement; and thereby require each and every child to attend some manner of instruction until such time as sufficient learning has been recognized, or level of chronological maturity reached. To insure that such required instruction is in and of itself valid and reliable, individual states have also mandated certain training and certification requirements for the classroom teacher. However, most states have failed to include equal protection for the student by requiring training related to computer literacy and the use of computers as an instructional device. The constitutional bodies of these states have therefore provided the potential for children to be exposed to the effects of computer-assisted instructional activities directed by unprepared and/or computer-illiterate teachers.

DANGER OF NON-LITERATE CAI INTERVENTION

Many professional organizations have evolved into controlling bodies in the areas of training, certification and monitoring. Medical professionals, members of the legal bar, psychologists and educators alike have recognized the inherent dangers in allowing individuals to practice certain crafts without requiring specific training and internship monitoring experiences. Yet while the average citizen will recognize the dangers of allowing an unqualified surgeon to take scalpel-in-hand within the operating room, many individuals do not understand the same potential for harm to the student within the classroom of the underprepared classroom instructor.

Because attendance of the child within a school program is mandatory, the parents can expect that education is a sine-qua-non necessity for future life and success as an adult. Because teachers must meet the certification standards of the particular state and/or district, parents can also expect that the giving of instruction is an achievement requiring (itself) some degree of training and preparation on the part of the teacher. And because the school day and school year is established by law to be of certain length, parents can also expect that whatever is mandated and required on the part of the teacher to teach these basic necessitites to the student, will be accomplished within that given timeframe.

The use of computer-assisted instruction (CAI) within the classroom will of course required the expenditure of some amount of time. If the time is well spent in terms of adequate learning and instructional material coverage, the child will be expected to progress in terms of intellectual as well as psychosocial development. If however, the student's time expended during the use of CAI does not produce the learning achievement which would have been reasonably expected had the student interacted with the teacher instead of the computer, then a problem of particular origin arises which must be evaluated and explained.

As with all instruction, expected (or purported) achievement on the part of the student is based not only upon the child's ability (viz., potential) to achieve, but also upon the structured use of the various activities of the classroom which purport to optimize such learning. Activity structure is based upon three indices of operation: content, the material being presented which is the expected equivalent of cognitive, affective and/or psychomoter understanding process, the manner in which the curriculum content is presented under the guidance of the instructor; and tooling, those mechanical devices which facilitate the presentation (process) of the material (content) which the student is required to learn.

Computer-assisted instructional techniques are therefore a process oriented methodology, allowed through the use of a tool known as a computer which provides the base for student learning and achievement of various curricular content. Utilized by a skilled professional, CAI has demonstrated marked increases in learning on the part of many children. In the hands of an unskilled professional however, the potential for great harm for the child is evident.

CAI software provides many opportunities to both the teacher and student for improved learning in terms of both effectiveness and efficiency. If however the teacher does not have the ability to recognize who should utilize computer-assisted instruction, and who should not; when it should be used, why, and how; and what characteristics of software learning packages are required to insure this learning, and how to determine the utility of one package over another -- the potential of a child expending limited time utilizing a technique which will not produce the necessary learning is obvious. And if this lack of learning achievement results in delayed progress as well as the inability to learn other concepts supported by prior learning, the student's ability to achieve at the level of expected comprehension is summarily precluded. Furthermore, if such lack of achievement results in the student being unable to progress towards some ultimate achievement which would have been reasonably attainable but for lack of progress attributable to mismanaged CAI intervention, then personal injury to the child has resulted.

WHAT IS TORT LIABILITY?

Tort may be defined as an injury or wrong done to someone for which civil action may be possible. Individuals held responsible for the content, process and tooling of the disputed action are said to be liable to tort. Either action or the result of the action may be addressable to liability, if the individual effected by such action can prove personal injury as a result of that action.

Three common issues of tort liability are often utilized to address specific details of the action in question.

The first, misfeasance, relates to the improper doing of an otherwise proper act. On the other hand, the failure to perform a required duty (especially by a public official) is called nonfeasance. Finally, the extreme of tort liability is defined malfeasance, the doing of an illegal act (wrongdoing) -- especially by a public official.

Three main questions are involved in determining personal liability as might relate to the operations and directives of the school environment. First, did the school employee owe a duty of care toward the plaintiff? Secondly, was there failure on the part of the employee to observe such a duty? Finally, was such a failure the direct, or proximate cause of the resulting injury?

WHY WOULD TORT LIABILITY APPLY TO CAI?

The use of computer-assisted instructional techniques to facilitate the ongoing activities of the classroom addresses each of the three indices of operation viz., content, process and tooling. The use of a computer would of course be defined as tooling. Specialized software in a particular curricular content area would be expected to display the attributes of specific instructional objectives, and would there define the operational index of content. Finally, the actual use of CAI methodologies as an alternate action to teacher-student interaction would be defined as process.

The first issue of tort liability, misfeasance, relates most directly to the operational index of process -- an alleged improper doing (CAI) of an otherwise proper act (teaching). This would be particularly acknowledged in the case of a student who acquired personal injury as a result of poorly arranged CAI sessions, due to the lack of knowledge on the part of the teacher as to how to utilize computer-assisted instructional methodologies.

The second issue of tort liability, nonfeasance, regards the operational index of content -- failure on the part of the teacher to perform a required duty (learning on the part of the student). Although it can be disputed that the teacher can not be held accountable for the result of the process of instruction, since the student may fail to learn regardless of the appropriateness of the methods employed by the teacher, it is reasonable to assume that the teacher is responsible for the appropriate content to support desired achievement. CAI software packages vary greatly on their ability to support desired instructional objectives of particular curricula. If the use of inappropriate or inadequate software for required instruction will demonstrate sufficient basis for the lack of desired learning achievement on the part of the student, potential liability is present.

The third and final issue of tort, malfeasance, relates only indirectly to the immediate defendant (teacher) in the demonstration of liability. Those individuals who are in a position to extend authority over the utilization of CAI in the classroom (principal, superintendent, state board of education, etc.) must assume responsibility for nonliterate CAI intervention. Although specific standards are part of the certification process of every teacher, few organizations have chosen to address the use of computers in the classroom as an instructional process and tool -- and therefore have failed to assure appropriate delivery of teaching services via computer by organizing general computer and CAI literacy inservices, and establishing certification standards regarding a teacher's use of CAI for instructional delivery.

ELEMENTS OF CAI TORT LIABILITY

In further examining the potential liability of schools related to a lack of student achievement which may be the result of computer-assisted instruction, seven elements of tort liability are identifiable. For the purpose of expostulation, each element will be summarily addressed as it applies to CAI tort liability.

Substance of personal injury. Can the student demonstrate that a priori expected achievement was denied as a result as inadequate learning via CAI? Such denied achievement might be defined as loss of scholarship, inability to proceed into the next higher-level curricular course due to inadequate skills, and/or the loss of ability to attain admittance to some organization or institution due to low achievement and/or entrance examination scores (e.g. college entrance). In this regard, personal injury should not be confused with claims of personal or psychological anguish as a concurrent result of alleged failure to perform as expected.

Duty of care. Can the student demonstrate that the school represents the interests of the state regarding the operationalization of the state mandate to provide mandatory instruction? That is, can the school be held responsible for providing services which it has been empowered to provide by legal obligation to the state? Furthermore, can the student show that the school recognizes the need for a duty of care, by demonstrating the school's requirement(s) for certification standards on the part of all classroom personnel?

Standard of care. Can the student demonstrate that because of his or her age and mental capability, the physical condition of the classroom, and the environment, the standards imposed upon the use of CAI were inadequate for operationalization of its use in the classroom? Can the student further show that knowledge of such deficiencies was a reasonable expectation of deficiency for perceived liability on the part of the school?

Reasonableness and prudence. With regard to teachers, the generally accepted standard of care would be that of a reasonably prudent teacher, not that of a reasonably, prudent layperson. Can the student demonstrate that the teacher did not do all things that were both reasonable and prudent in determining the use of computer-assisted activities within the classroom instructional environment? Can the student further show that in determining similar learning activities for CAI presentation as opposed to 'normal' presentation, the teacher did not employ reasonable and prudent standards of evaluation that should have been expected?

Proximate Cause. Can the student demonstrate that CAI as an action produced an event, and therefore without which, the event could not have happened; and that furthermore, this event is responsible for the occurrence of personal injury? Proximate or legal cause is the sequential connection between the individual's negligent conduct and the resulting injury to another person.

Feasance. As performance of a condition or duty, can the student demonstate that there was failure to act (i.e. nonfeasance), good intentions but conduct falling below an established standard (i.e. misfeasance), or deliberate injury (i.e. malfeasance)?

Culpability. And finally, can the student demonstrate that it was not reasonable to expect the student to recognize in advance the potential shortcomings of the inadequate CAI system in use? and therefore, the potential harmful effects which might be visited upon the student's future?

UNCONTROLLED CAI UTILIZATION NEGLIGENCE

There exist four elements of negligence in the determination of liability for improper, inadequate or inappropriate action(s): (1) duty to the individual in question; (2) standard of care for the performance of such a duty; (3) the proximate or legal cause of the alleged personal injury; and, (4) the personal injury itself, or actual loss.

The primary question to be answered, is often that of the foreseeability of the injury -- that is, was there reasonable expectation that such an injury might occur based upon previous experience or knowledge? As in all professions, the actions of professional endeavors have been researched, assessed and evaluated for their validity and reliability in addressing the main issue at hand. That the introduction of computer technology as an instructional device within the classroom could take place without the appropriate training of the teacher, and the adequate evaluation of the software for instruction, provides the undeniable foundation for alleged negligence on the part of the educational establishment.

STANDARD OF CARE FOR THE USE OF CAI

The expostulatory nature of this paper signals new directions for the leaders of education regarding the introduction of instructional technology into the learning environment of the classroom. In summary, three specific issues must be addressed:

1. How to best utilize the computer as a facilitator of the instructional and learning process;

2. How to recognize and determine the relevance of various software packages to the established instructional objectives of the curriculum in question; and,

3. How to most effectively train the teacher as instructional leader to optimally operationalize the basic tenets of both (1) and (2) above.

The case of computer literacy training is clear -- at both the undergraduate university level as students finish their initial training for certification as well as at the building and district level as teachers participate in structured inservice activities -- if computer-assisted instruction is to contribute to the positive efforts of the learning environment, the teacher must be trained as a professional to insure such contribution.

Without proper and professional training, the teacher-user of CAI violates the statutes of the instructional code, promotes the infusion of innappropriate practice into the learning environment of the child, and sets the stage for the eventual preclusion of future, valid instructional technology as a reliable facilitator of desired curricular presentation and teaching.

Without proper and professional training in the foundations and concepts of computer and CAI literacy, the teacher is liable for for resultant personal injury to the child.

THE SELF-SUFFICIENCY APPROACH TO COMPUTER LITERACY

Brent E. Wholeben, Ph.D., Associate Professor
Department of Educational Administration and Supervision
The University of Texas at El Paso (El Paso, Texas 79968)

The computer literacy program under the self-sufficiency framework consists of 16 distinct though related components (or activities) organized within a six stage developmental framework.

DEVELOPMENTAL STAGES

The self-sufficiency approach allows for maximum faculty and staff participation over an approximate 7-month implementation period. To provide as optimal an instructional environment as possible for structured, developmental learning on the part of all participants, the program has been organized into six developmental stages or modules, of 2-4 activity components each. These stages are identified as:

Stage I: Awareness (providing the basis for an assessment of pertinent programmatic and institutional needs, and the foundation for the upcoming inservicing of select faculty (viz., cadre);

Stage II: Initial Literacy (establishing a core of computer literacy knowledge for both cadre and general faculty/staff, and the rudiments for unstanding potential microcomputer applications in education);

Stage III: Cadre Training (presenting a series of instructional objectives related to computer-assisted instruction (CAI) and management (CMI) to select faculty, and the background for transmitting these same instructional objectives to other faculty and staff);

Stage IV: Administrative Inservice (offering the necessary introductory training on specific, office-associated software to management staff, and the initial incorporation of this software-application within current school administrative functioning);

Stage V: Facilitator Inservice (formalizing suggested guidelines for participation by the cadre in the further developmental needs of the general faculty, and the initial design for continuing computer literacy instruction for the upcoming academic year); and,

Stage VI: Faculty Inservice (executing a series of specific computer literacy sessions based upon demonstrated (specific) needs of the general faculty, and the finalization of suggested literacy components for the upcoming academic year).

ACTIVITY COMPONENTS

Each of the identified developmental stages contains specific components or activities which delineate the individual awareness, literacy, training and inservice components described above. These activities range from a frequency of 2-4 per stage, and denote the individual tasks which will be accomplished in support of the particular stage addressed.

Awareness Components

The Pre-Needs Assessment Consultation Component. Initially, meetings will be required with the identified cadre faculty ("select-group") who will act as internal trainers and resource people to remaining faculty and students. The cadre will be responsible for eliciting information from other faculty as to current interest, knowledge and need related to computer applications for educational functioning, and for determining a list of priorities for designing the subsequent computer literacy needs assessment strategy for execution. Organizational patterns will also be developed regarding liaison coordination with school administration; and the development of periodic status reports and/or updates.

The Select-Group Computer Literacy Awareness Component. The identified cadre of faculty will be involved in preliminary computer literacy awareness session(s) so as to provide a knowledge foundation prior to the development of a valid and reliable needs assessment instrument. The topics for this general instructional component will include certain hands-on activities as well as in-depth discussions of the microcomputer's potential for instructional implementation. A variety of available CAI/CMI software (courseware) packages will be demonstrated for increased understanding. In addition, select management applications for classroom recordkeeping will also be utilized to show the range of the computer's potential to the classroom teacher.

The Full Needs Assessment And Information Analysis Component. A formal instrument will be directed to all faculty and staff residents, seeking individualized and collective information regarding specific needs which currently exist as well as future requirements and suspected potential. Data will also be gathered regarding collective faculty and staff experience, and a general view of their preponderance either towards or away from the use of computerized technology within the educational setting -- whether instructional or administrative. Individual interviews will be solicited from interested faculty for more specific (both subjective and objective) information associated with technological interventions for classroom use; and a series of group discussions may be scheduled based upon time and interest. The results of this assessment will provide the proper foundation not only for tailoring a general-group component of computer literacy inservice, but also for projecting specific instructional needs of the cadre (select-group) during their literacy training sessions.

Initial Literacy Component

The Select-Group Preservice Computer Literacy Component. Described as a preservice rather than inservice component, this activity will prepare cadre members to act as resource personnel for the upcoming general-group inservice computer literacy component. This activity will prepare each cadre member to serve as a trainer and station-director during the general literacy session(s); and will furthermore provide additional experiences to cadre faculty in the areas of problem-identification and solution-formulation regarding computer applications to the classroom environment. The particular preservice component is also intended to serve as a 'part two' or follow-up to the select-group literacy awareness component.

The General-Group Inservice Computer Literacy Component. Many of the tasks associated with the select-group awareness component will be re-administered in this general-orientation session. Cadre members will serve as co-presentors; and will provide station-oriented inservice demonstrations for general faculty and staff member instruction. In addition to the major goal of literacy and awareness instruction, this component will serve to introduce the self-sufficiency computer literacy program to general faculty and staff; and to prepare the faculty to potential changes within the instructional climate for the future.

SELF-SUFFICIENCY COMPUTER LITERACY APPROACH (Wholeben,1983) Continued

Cadre Training Components

The Select-Group Computer Literacy Training Component I. This first of three training components will be comprised solely of an introduction to the BASIC programming language, APPLESOFT; and its use on the APPLE II PLUS DOS 3.3 operating system. Rudiments of the language's nomenclature and syntax will be explored; and sample programs will be written and executed on the microcomputer. Concepts of disk storage and retrieval of programs will be presented; and the utility of program editting functions discussed.

The Select-Group Computer Literacy Training Component II. This second component will serve to present advanced concepts of APPLESOFT BASIC language programming to the cadre members; and will include the utility of disk-access from within programs for information and scratch-file swapping. Advanced techniques for CAI/CMI development and implementation will be introduced via the use of the APPLE PILOT software package; and individual projects will be planned for the incorporation of a CAI-PILOT courseware session within each cadre member's classroom setting.

The Select-Group Computer Literacy Training Component III. This last in the series of three training components for cadre faculty will be concerned with the survey techniques required in the evaluation of market-available courseware for instructional-classroom use. Sample courseware packages will be evaluated under a specified list of criterion references; and in turn, a plan for acquisition of software preliminarily designed. Finally, a strategy for gaining additional information concerning market-available software will be formalized for administrative approval.

Administrative Inservice Component

The Office-Group Inservice Computer Literacy Component I. This first of four inservice components for business and office staff will be comprised solely of an introduction to the use of the CONTROLLER software package for business transactions requiring general ledger, accounts payable and accounts receivable recordkeeping. Strategies for incorporating this business software into the ongoing operation of the School will be designed; and the rudiments of building the requisite data base will developed.

The Office-Group Inservice Computer Literacy Component II. This second activity component will survey the utility of the DB-MASTER data base management system for general school recordkeeping and file archival requirements. An introduction to the software package will also include the beginning aspects of developing the requiring data base; and the transferral of physical records to a disk-storage environment.

The Office-Group Inservice Computer Literacy Component III. The use of the APPLE II PLUS microcomputer will be demonstrated for its utility as a software supported word processing machine. With the APPLEWRITER II software package staff will be trained to develop correspondence formats, mailing labels, and general manuscript editing skills via an electronic, screen-oriented worksheet rather than the usual typewriter-machine copy.

The Office-Group Inservice Computer Literacy Component IV. This last in the series of four inservice components for business and office applications will be concerned with the use of the VISICALC software package for educational evaluation and decision-making. Such uses as enrollment projections, fiscal auditing and financial projections, and program evaluation will be demonstrated.

SELF-SUFFICIENCY COMPUTER LITERACY APPROACH (Wholeben, 1983) Continued

Facilitator Inservice Component

The Select-Group Computer Literacy Facilitator Component. A variety of requirements will be satisfied under the genus of this particular activity. Cadre members will be trained to act as internal resource personnel for the upcoming academic year. Procedures and strategies for optimizing use of these trained individuals will be formalized and forwarded to school administration for input, modification, and subsequent approval. Certain advanced applications will be discussed as appropriate and/or demanded. Finally, the structure for the upcoming three structured-demand literacy components for general faculty will be designed and programmed for implementation. As with the general-group literacy session preceding, cadre members will serve key instructional functions in the ensuing structured-demand sessions.

Faculty Inservice Component

The Structured-Demand Instructional Literacy Component I. This first of three proposed training sessions will be structured around prioritized needs on the part of faculty members concerning CAI/CMI incorporation requirements. The topic(s) will be progressively identified throughout the implementation of the preceding developmental stages (and their corresponding components).

The Structured-Demand Instructional Literacy Component II. This second session will continue to provide specific, in-depth training for interested faculty.

The Structured-Demand Instructional Literacy Component III. This last in the series of three training sessions will conclude the school computer literacy program for the given year.

TASK DELINEATION OF COMPONENT ACTIVITIES

Each of the above sixteen activities will involve several generic requirements related to its reliable implementation. These areas have been identified as follows, and will be utilized to describe general efforts over-time in the successful implementation of these activities:

1. Consultation, those efforts and requirements which will promote the best communication, liaison decisioning control during the course of this proposed project;

2. Planning, those activities and tasks which will detail all proposed, intended efforts for implementation as formal program design and execution;

3. Materials Development, those items which are utilized for instruction, training demonstration and model building which provide the bases for predictable learning and method incorporation;

4. Simulation, those experiential episodes which provide initial piloting of future behaviors for evaluation and improvement;

5. Implementation, those activities which denote the actual execution of a proposed program of functioning;

6. Evaluation, those measures of effectiveness, efficiency and satisfaction which allow feed-back to program developers for improved future implementations; and

7. Follow-Up, those efforts for determining the retention and personal incorporation of materials within actual classroom and office functioning; and those summative measures of overall computer literacy program utility.

NEEDED: LESS COMPUTER APPREHENSION
WANTED: COMPUTER ENTHUSIASTS
FOUND: IN-SERVICE TECHNIQUES

Connie Carroll Widmer
Northern Kentucky University
Highland Heights, KY 41076

Janet Parker
University of Louisville
Louisville, KY 40292

Despite the laudatory publicity given microcomputers, surveys show that not all teachers are welcoming them with enthusiasm. A few teachers outrightly reject them; others are apprehensive about computers and confused as to how computers will affect them, their jobs, their classrooms.

To meet both the affective and cognitive needs of these teachers, initial in-service experiences must be carefully planned. Based on the research and experience of the presenters, this paper discusses problems and issues related to providing effective in-service training and presents specific guidelines.

The effects of anxiety upon learning are familiar to all of us. A certain degree of anxiety can provide the necessary "shot in the arm." Beyond that critical point, however, a "Catch 22" situation ensues: the learner's eagerness to master a subject can create anxiety which, in turn, impedes mastery.

Microcomputer Anxiety

The arrival of the microcomputer on the educational scene (estimates are that over 100,000 microcomputers are now being used in education in the United States (Moursund, 1982, p. 6)) has opened new vistas of learning---but also created a new source of anxiety. While computer advocates triumphantly proclaim a new era for schools and herald the increased powers available to us, many teachers, trained before the dawning of the microcomputer age, are facing this newest challenge with less than unadulterated enthusiasm. A study (Lichtman, 1979) comparing educators' attitudes toward computers to those of the general public found that "in general, educators seemed less enthusiastic about the computer's role in society than did the general public." Teachers seemed more convinced than the public that computers dehumanize society (55% vs 37%), less sure that computers will improve education (64% vs 87%) and more concerned that computers will isolate people from social interaction (30% vs 19%). Sixteen percent of the teachers were disturbed at the prospect of losing their jobs to a computer.

Further studies (Stevens, 1980, 1982; Parker and Hazuga, 1982) compared attitudes of teachers, teacher educators, and student teachers. Stevens found that all three groups were significantly more knowledgeable about computers in the later study than were similar groups in the earlier study; however, teachers and teacher educators exhibited no corresponding change in attitudes toward computers in education. Attitude scores of student teachers, however, were significantly higher in the later group. Parker and Hazuga, on the contrary, found the student teacher subgroup to be the most anxious, despite the fact that this group had received the most training on computers.

These studies document what many of us have witnessed: teacher apprehension about computers is not uncommon. For some, the technology of computers is intimidating; the world of silicon chips, ROMS and RAMS, cables and circuits, seems complex and far removed from the hope for a convenient teaching aid. Other teachers regard the microcomputer as just one more fad of the moment, which, like others, will fail to deliver the promised performance. Like other fads, they tell themselves (hopefully, at times), this, too, will pass into oblivion---or into a storage room---if they just wait long enough.

Perhaps the greatest worry of all is the fear of embarrassment---of not knowing how to get started while youngsters are blithely punching away at the keyboard. Many fear making mistakes which will appear on the monitor for the world to see.

Whatever the fears, realistic or exaggerated, based on misconceptions or on a simple lack of experience, they are real, and, unless faced and dispelled, they can prevent teachers from becoming computer literate and comfortable.

Effective In-service Training

Since positive attitudes toward computers are essential for the implementation of successful computer programs in the schools, any steps that can be taken to inculcate these attitudes promise to pay large dividends. In providing either short-term in-service workshops for teachers, or longer credit courses, attention must be given to the approach as well as to the subject matter. The suggestions that follow are designed to promote enthusiasm and decrease apprehension, while providing a solid introduction to microcomputer use in education.

1. Since much anxiety is rooted in the unknown, have teachers become friends with the computers as quickly as possible. Hands-on experience should precede technical explanations. A few simple directions can enable a novice to boot the system and interact with a user-friendly program. Thus empowered to control the computer, and assured that it is extremely unlikely that they will damage anything, participants begin to enjoy the experience---and wonder why they were so hesitant to take the plunge into the computer realm.

2. Nothing decreases anxiety like success. Therefore, it is essential to begin in-service training with experiences in which the participant will be successful. In addition to choosing exemplary software, this also means that beginning programming experiences should be carefully designed to lead to successful outcomes. Demonstrating the use of the computer as a calculator, followed by easily interpreted statements, such as the BASIC commands PRINT, and GO TO increase the user's sense of power over the machine and provide interesting outcomes.

3. Provide program examples that are educationally relevant and pedagogically sound. Programs are available which simulate the interactions in a food chain, show the positions of stars on any given night of the year, allow the user to manipulate the factors operative in an electoral campaign, and provide for accident-free titrations in a chemistry lab. Tutorials will help to teach addition of fractions, solving of algebraic equations, reading a micrometer, and correct usage of parts of speech. Drill and practice programs abound; the better ones give the user a choice of level of difficulty, provide instant reinforcement, correct wrong answers, keep score, and use graphics to provide a bit of flamboyant elegance. It should frequently be reiterated that these are not programs which teachers *should* use, but ones which they may *consider* using. The ultimate decisions as to what is appropriate for a given class rests with the teacher.

4. Two at a computer seems to be the ideal number. Partners can discuss what they're doing, aid each other in problem-solving, detect each other's errors (often typing mistakes) and offset the tension which can come from working alone.

5. Help the class to identify comparisons and contrasts between teaching

a topic via computer and other instructional methods. Demonstrate potential computer advantages for certain topics and tasks. Prepared lesson plans for a specific topic which incorporate computer use and provide preparatory and follow-up materials serve as an additional prod to teachers to actually try a computer-based lesson in their classrooms.

6. The time-saving aspect of data management computer programs is very attractive to teachers. Programs for record keeping, gradebooks, and diagnosis and individualization procedures should be demonstrated and discussed.

7. Discuss different logistical arrangements for the classroom. One computer with a large monitor may be all that is available in many classrooms. Some schools have a sign-up system for a portable microcomputer unit. Microcomputer labs offer yet another possibility. Each situation demands a different mode of instruction and different skills on the part of the teacher. If the day comes, as Seymour Papert (1980) advocates, when each child has his or her own microcomputer for school use, we will have to drastically rethink what a classroom is supposed to be.

8. Computers have become a great leveler with respect to the learning process. Adults do not possess the advantage of having acquired a life-time of experience with computers before beginning to teach about them. If such a situation arises (and it probably will) admit your inability to explain what has happened or to answer a specific question. Classroom teachers will face students who have been at the keyboard long before they were. The trainers of teachers can set a good example by admitting that the field is vast and by doing the necessary study and research to pursue questions which arise.

9. Face directly any fears that computers may take over the classroom. Holmes (1982) raises directly the threatening question: "Can machines perform all, or even most, of the teacher's functions?" A discussion among participants of which functions the computer could perform and which tasks they would like it to perform, along with an explicit recognition of those things a teacher can do that computers will never be able to do, should do much to alleviate this particular fear.

10. There has been much debate over the meaning of computer literacy. A considerable portion of this debate centers around the degree to which computer literacy should be identified with knowledge of programming. There is no doubt that it is advantageous for teachers to know some programming. This enables them to better understand the process by which a computer functions and also gives them the ability to make minor changes in accessible programs. However, it should be kept in mind that the vast majority of teachers will use pre-programmed materials in their classrooms. What is most essential for teachers is skill in evaluating these materials and incorporating them into their teaching. Just as most of us do some writing, but do not write the textbooks we use, so most teachers should be able to write a simple program, but will not write most programs that they use in the classroom.

The above constitute a few practical suggestions for implementing effective in-service computer training for teachers. Such training focuses on providing useful content in a non-threatening atmosphere, replete with interesting, useful, successful computer experiences. Such experiences should do much toward increasing the supply of knowledgeable, enthusiastic, computer-using teachers.

REFERENCES

Holmes, G. Computer-Assisted Instruction: A Discussion of Some of the Issues for Would-Be Implementors. Educational Technology, 1982, 22, 7-13.

Lazarus, M. So Complicated, They're Simple: A Reassuring Word on Classroom Computers. Principal, 1982, 62(2), 39-41.

Lichtman, D. Survey of Educator's Attitudes Toward Computers. Creative Computing, 1979, 5, 48-50.

Moursund, D. Teacher's Guide to Computers in the Elementary School. Eugene, OR: International Council for Computers in Education, 1982.

Papert, S. Mindstorms. New York: Basic Books, 1980.

Parker, J. and Hazuga, M. Educators' Attitudes and Knowledge of Computers in Jefferson County. Unpublished paper, University of Louisville, 1982.

Stevens, D. J. How Educators Perceive Computers in the Classroom. AEDS Journal, Spring, 1980, 221-232.

Stevens, D. J. Educators' Perception of Computers in Education: 1979 and 1981. AEDS Journal, Fall, 1982, 1-15.

INSTRUCTIONAL COMPUTING IN THREE SOUTHEASTERN STATES

Jane H. Wilde
Voorhees College
Denmark, South Carolina 29042

ABSTRACT

Computer Literacy and instructional uses of the computer are increasingly significant in higher education. However, instructional computing has appeared to be unevenly spread throughout the nation: women, minorities, rural areas and the southeast have been significantly less likely to have access to computer assisted instruction and to programs of computer literacy. This paper compares the educational uses of the computer in three southeastern states with emphasis on South Carolina. It also discusses plans underway to promote computer literacy and to improve instructional uses of the computer.

This survey focuses on the facilities for instructional computing in three southeastern states: South Carolina, Alabama and Mississippi. These states primarily rural, often rank near the bottom in national statistics relating to education, and all have large minority populations. Recent studies have shown that computer experience is not equally distributed across the national population and that women, minorities, persons in rural areas and the southeast have been less likely to have had any experience with the computer. The survey is an effort to determine what exposure college students and faculties were having to the computer.

One could assume that there would be substantial development in areas traditionally identified with computer usage, such as business, mathematics, the sciences and computer science itself. But what about the application of computers in the humanities and other liberal arts areas? What about the increasingly critical need to develop college graduates who are computer literate? Are there general requirements for all students in computer literacy in order to graduate? What efforts are being made in faculty development? To what extent is computer assisted instruction being utilized in all areas of instruction and as a part of major requirements? This survey has attempted to answer some of these questions. This paper will highlight some of the major conclusions. (An updated and fuller interpretation will be distributed at the meeting).

The degree of interest in instructional computing is shown by the fact that of the initial 153 colleges surveyed, 61%, or 93, returned the surveys on the first request. A follow-up mailing is expected to bring more responses. Approximately 35% of the respondents also wished to receive copies of the tabulated results.

In terms of the size of the institutions surveyed who reported enrollment, 10 had an enrollment of over 5,000. 46 had under 2,000 students. Not all colleges could identify whether their clientele were primarily from rural areas. South Carolina and Mississippi showed a preponderance of students from rural areas as opposed to Alabama, which showed an even split.

Institutional type included the following:

	S.C.	MS.	AL.
Public-4 year	12	3	8
Public-2 year	12	10	16
Private-4 year	12	7	6
Private-2 year	3	2	1

Many institutions both two and four year indicated a computer science major or a degree program in computer science was available.

Computer science majors are available primarily from the state supported four and two year colleges. The survey indicated that the two year college systems in the three states do offer degree programs in computer science. Some of these are transfer programs but others are designed to provide the necessary training for positions in business and industry.

MAJORS	S.C.	MS.	AL.
Public-4 year	7	3	6
Public-2 year	8	6	8
Private-4 year	2	0	1
Private-2 year	1	0	0
Women's	0	0	1
Black-4 year/Pub.	1	1	0
Black-2 year/Pub.	1	0	1
Black-4 year/Ch.	3	0	2
Church related	3	2	2

A number of institutions offer minors only.

Computer assisted instruction is in use mainly in the business and science areas. However, a number of institutions have found applications in fine arts, humanities, health, and other areas.

Institutions indicating CAI usage total 16 in South Carolina, 13 in Mississippi, and 16 in Alabama. One institution responding indicated it had abandoned CAI. Fifteen institutions indicated that they were using CAI in the basic skills areas and at least two indicated they were starting a program. One indicated the use of CAI in Music for aural training, another in teaching Spanish. Nine institutions reported using CAI in English language and five in English literature. Eight reported using CAI in Education, which is a key area in many respects because the increasing use of information technology will require elementary and secondary school teachers who are computer literate.

SOURCES OF PROGRAMS	S.C.	MS.	AL.
Faculty Developed	17	10	16
PLATO	3	2	1
SIGI	1	2	1
Other Commercial	8	6	3

There is a strong argument for faculty developed programs which are tailored to meet the special needs and characteristics of students on individual campuses, particularly in CAI. Faculty should also be aware of the availability of faculty developed programs from other colleges which are disseminated through CONDUIT and a number of other clearing houses.

One of the goals is the development of computer literacy for all students obtaining a terminal degree. Few colleges in the survey have a computer literacy requirement for graduation although many students do gain an exposure to the role of computers in today's society and the ability to use them. In South Carolina two colleges indicated that they had a computer literacy requirement and one was planning one for the future. One college in Alabama indicated there was a computer literacy requirement and there were none in Mississippi. However, the response to the question "do you have any courses designed to provide computer literacy as part of general requirements" produced a variety of answers, some of which showed that, while there was no general requirement for all students, there were a variety of options open to provide computer literacy in some programs. One in South Carolina stated that they had started such a course as an elective this year; another had a unit on computers in the general education math course. A number of colleges indicated that they had an introduction to computer science course or an introductory data processing course that would provide an introduction to computer literacy. 16 of the South Carolina colleges indicated they had such a course. In Mississippi, 8 colleges indicated the existence of a course. One reported a freshman interdisciplinary course which included English, Math, and an introduction to computers, while another indicated there was a course on an elective basis. Twelve Alabama colleges showed some type of computer literacy course.

Another course related to plans for developing a computer science major. In South Carolina, 2 colleges are involved in a feasibility study indicated a major would be available within 2 to 3 years; 2 were slated to start in fall of 1983. One college in Mississippi had projected plans for a major, while in Alabama two colleges were purchasing equipment to begin a major and one was starting the major in the fall of 1983.

A substantial number of colleges reported a variety of activities in faculty development related computers and the development or expansion of computer assisted instruction. Certainly one of the major problems in implementing a program is to overcome computer anxiety on the part of instructors who have little knowledge of computers and no hands-on experience as opposed to an increasing number of students who have been initiated to computers either in pre-college education or by the burdeoning computer games. In South Carolina, 19 institutions reported faculty development activities in this area, as did 13 in Mississippi and 19 in Alabama. There was wide variety in the sorts of activities provided. Some were short courses, some provided tuittion assistance or free tuition for faculty to take computer science courses. Others provide inservice workshops and seminars. One utilized an in-house developed computer literacy manual. It was interesting to note that in some of the Mississippi institutions there were faculty development programs in this area in spite of the lack of much of any programs in these same colleges. This would seem to indicate the laying of a foundation for eventual expansion of CAI.

How were on-campus computers used? One of the problems that has been cited is a tendency for one mainframe computer or mini computer that is shared to be monopolized by administrative uses. In many instances the data indicated that the same computer was used for administrative and instructional uses, but no proportions of time were given. The respondents indicated that computers for instructional purposes were used as follows:

	S.C.	MS.	AL.
Instruction Only	15	5	7
CAI	8	8	9
C.S. Major	12	9	14
Admin & Ed. Use	29	18	19

A wide variety of computers were in use in the three states.

A survey produced the following figures on qualified faculty in the computer programs offered.

In South Carolina, 5 colleges reported faculty with doctorates in computer science, one with 13, another with 7, and one each with 2, 4, and 1. In addition, one had 3 doctorates with extensive computer background and another had one. 14 colleges reported having two instructors with masters in computer science, one reported 6, another 4, and five reported having one. Two reported having one instructor each with a bachelors degree in computer science. In addition, a number of colleges reported having faculty with computer training as follows: one with six instructors, one with 1.5, 10 with one. The other states followed similar patterns. 16 Alabama institutions reported a total of 57 master's in computer science, 5 reported 13 doctorates, and 15 institutions reported 93 with training in computer science. In Mississippi, 4 institutions reported a total of 17 doctorates, 13 reported having 27 masters, 2 reported 3 instructors with bachelors in computer science, and 11 reported 26 with training in computer science. These figures amply illustrate the shortage of qualified instructors in computer science and computer assisted instruction in higher education. In all three states, colleges reported having doctorates in fields such as math or chemistry or management who had an extensive computer science background.

It was interesting to note that few of the institutions responding were members of a consortium for instructional computing. A few indicated membership in the Data Processors Management Association. The responses would also indicate that there are no statewide associations for instructional computing. However, the technical colleges in South Carolina indicate that they had an informal network for sharing information. In addition, some of the black colleges have been involved in the NSF funded Mid Atlantic Consortium on Educational Computing. However, this grant is coming to an end. It would seem useful for more permanent networks, state chapters of AEDS, or other instructional computing grants to develop in these states to share ideas and to promote a wider use of instructional computing.

In summary, there appears to be an increasing development in instructional computing in these states. It includes a wide variety of faculty development activities on many campuses. However, it does not yet appear that there is a concerted move toward developing a general computer literacy requirement for all students. It would appear that the need for expansion in this area is gaining recognition. However, the current economic climate and retrenchment in funding affecting both public and private institutions presents an obstacle. The rapid development of new, better, and less expensive hardware, however, is a brighter aspect in the growth of instructional computing.

MICROCOMPUTERS IN GEOGRAPHICALLY ISOLATED SCHOOLS

Ray Wilson
Fort Lewis College
Durango, Colorado 81301

ABSTRACT

Few articles about CAI or computer literacy's place in the K-12 curriculum have focused on geographically isolated schools. One such region with interesting developments is the southwestern corner of Colorado. Districts of less than 600 students predominate, even though their boundaries cover vast geographical areas.

While developments in the classroom uses of microcomputers mirror the patterns in urban and suburban districts to some extent, many unique approaches are developing in this region. Involvements by teachers, students, administrators, professors from higher education, and a wide variety of community people are creating an unusually fertile ground for microcomputer developments.

Southwestern Colorado is a land of natural beauty, varied outdoor recreation, tourism, mineral resources and few residents. These residents are geographically isolated from the rest of the state by natural and psychological barriers. With the legislature dominated by elected representatives from the population centers east of the mountains, the entire region west of the Continental Divide, (referred to as the Western Slope,) frequently feels that its problems are ignored, serving only as a water and recreation resource for Eastern Slope residents. These problems are magnified in the southwestern corner where Red Mountain Pass (11,018 feet) and Wolf Creek Pass (10,850 feet), the main highway routes into the area from the north and east, respectively, are frequently closed in the winter months due to avalanches or 'white-out' conditions. As in other parts of the country with similar challenges, a sense of independence exists in the psyches of many of the residents.

The largest city in this region is Durango with a population of 12,000. Durango is the center of tourism, boasting a narrow gauge railroad, skiing, and big-game hunting among its attributes. Other cities and towns in the area are probably not household words in other parts of our country. Towns such as Dove Creek, Cortez, Dolores, Mancos, Silverton, Bayfield, Ignacio and Pagosa Springs probably do not stimulate recognition in our minds. Hispanics and Native Americans comprise significant portions of the population. Such communities would not generally be expected to be in the forefront of technical advancement.

Cable and satellite TV have helped link the region with the outside world, but many communities still get poor reception from one or two TV channels from Albuquerque or Salt Lake City which are broadcast via community translators. Other residents receive no TV; even radio broadcasts are limited by the terrain.

Financial resources for the public schools are scarce. Mesa Verde National Park, Hovenweep National Monument, Ute Mountain Ute and Southern Ute Indian Reservations, San Juan National Forest and other public lands provide no tax base. Funds provided from the state are not enough to offset the added expenses of operating schools in districts of low population density, e.g., some students must ride school busses more than two hours per day. Many districts have total enrollments of less than 600.

This setting might appear to be barren ground for rapid development of microcomputer resources. However, the opposite is the case. A consciousness of the current state-of-the-art in instructional computing has developed, and these communities are devouring information available about the application of microcomputers. This is true of educators, business people, farmers and ranchers. This paper will briefly describe the contribution of various people and organizations to the development of this computer consciousness.

Overview of Microcomputer Applications

Most microcomputers have been acquired by the schools since the fall of 1981. Although a few microcomputers were in use before that, almost all development has taken place in the last two school years. No attempt will be made to compare ratios of microcomputers to students. Any ratios cited will probably be outdated before this paper is presented.

Space does not permit a description of everything being accomplished with microcomputers in this region. Instead, a few districts will serve to illustrate the scope of development taking place.

Ignacio

Students in the Ignacio schools are almost evenly divided among Anglos, Hispanics and Native Americans, primarily Southern Utes. Farming and ranching are predominant occupations. The district has one elementary school, one junior high school, and one senior high school with a total of 900 students. Computer literacy and programming are emphasized at the high school. The program was developed by a mathematics teacher and coach who has experience as a programmer. He is also teaching graduate level courses in computer literacy and programming for other teachers from all levels in the district. A frequent topic in the faculty lounges is the progress or problem in completing a programming assignment.

Word processing and computer applications have become part of the business education curriculum. Enthusiasm from the business education teacher and students caused the school board to purchase a computer with word processing package for the central administrative office. Previously, school business records have been handled as they had been for the past half century.

CAI has been used to some degree, but not extensively. Plans call for the acquisition of additional computers with a significant use of CAI in the elementary school.

Silverton

Silverton, nestled in a valley at 9,300 feet, is the most isolated community. One highway goes through Silverton; it is frequently closed by avalanches, blocking access from both the north and south. The economic health of Silverton rises and falls with the gold and silver market. Approximately 160 students (K-12) attend classes in the single school building.

Leadership from the business education teacher provided the catalyst for microcomputing. CAI, computer literacy and programming are being taught. A workshop during the fall of 1982 for graduate credit provided an introduction into these areas for other teachers and substitutes. Another workshop is currently underway for the other teachers in the school.

Enthusiasm from the teachers and students has been so great that other townspeople have 'demanded' a concurrent workshop of their own.

Additional leadership is being provided by the music teacher. Never having used a computer, he attended the fall workshop; actually, 'devoured' is a better word than 'attended.' He has become a self-taught expert in Apple graphics and animation. He and a student synchronized a musical number with computer graphics for the Christmas program, only three months after his introduction to the Apple II+.

Dove Creek

Dove Creek calls itself the "Pinto Bean Capital of the World." Located on a windswept plateau near the base of the mountains, the schools enroll less than 350 students (K-12). A unique blend of school and community cooperation expanded their microcomputer resources. A community grant purchased two microcomputers to go along with two others purchased by the school district. The community grant stipulates that the microcomputers be available to community people after 1:00 p.m. daily. Scheduling of computer time has become difficult due to the heavy demand.

A workshop was conducted by the author on Friday nights and Saturdays during the spring of 1981 for an assorted group of 44 teachers, counselors, school district secretaries, farmers, ranchers, and business people. Another 30 people are currently taking the beginning workshop while 20 of last year's group are doing advanced work. The fervor of this community to perfect their microcomputer skills is unsurpassed.

Durango

The school board is considering making computer literacy a graduation requirement. A district-wide committee is considering computer literacy as a K-12 program. Computer literacy and programming are the current primary thrusts of the high school (approximately 850 students). The library at the high school has access to the reference resources of The Source. Special education students use CAI to build and reinforce basic skills and life skills such as personal budgeting and finance.

CAI is being used by individual teachers, but has not made major inroads throughout the district. The teachers involved, however, are beginning to attract the positive attention of other teachers in their schools. Major obstacles include some administrators who feel computers are a passing educational fad and prefer schools to be 'human' not 'electronic.' Adult education classes and workshops for teachers are filled to capacity. The public library has two microcomputers available for a nominal hourly rental charge in addition to access capabilities to several on-line databases.

Board of Cooperative Services

The San Juan Board of Cooperative Services (BOCS) provides a variety of shared services for five districts. A software library is maintained at the BOCS office. Software may be checked out by any teacher of a member district. Other services include minor equipment repairs, computerized storage of special education records and consultation on problems encountered in the use of microcomputers in the schools. These services are due to the interest and leadership of the BOCS director.

Fort Lewis College

Fort Lewis College became the first college or university in Colorado to require coursework in the use of software for instructional purposes as a requirement for teacher certification. Student teachers are using software resources in the classes they teach. Classroom management simulations on the college computer system are required for teacher education students. Tutorial software and courseware are being developed for the fundamentals of classroom test construction. Some pre-service teachers have taught mini-courses for elementary school gifted/talented students. Parents of these elementary students enthusiastically report that the mini-courses have been the most educationally motivating experience in their children's lives.

A three week workshop for senior citizens about home computers was held during the summer of 1982; because of high demand, two similar workshops will be held during the summer of 1983.

Many other activities are being planned. An introduction to microcomputers for alumni is being considered as a pre-homecoming activity. A computer camp for students and their parents is being developed. The combination of microcomputers and the summer outdoor recreation possibilities create the opportunity for a well-rounded camp experience.

Conclusion

The districts and institutions described in this paper are only part of the microcomputer explosion experienced by schools and communities of southwestern Colorado. This explosion was not preceded by a needs assessment, development of a program to meet those needs, and training of teachers to provide a systematic implementation of the program. Rather, the explosion has been led by people who have sensed the magical excitement of the world of the microcomputer and want to get involved.

The author was originally skeptical that a cross-section of teachers, students, farmers, ranchers, business people, secretaries and housewives could become computer literate and stimulate each other to higher levels of achievement. He is no longer skeptical; it is happening repeatedly in this area. Some have accused him of being a 'pusher' to create microcomputer addicts. There also seem to be pushers in every school district.

The argument could be made that this spontaneous growth is costly, both in dollars and inefficient use of time. Granted, a case could be made that planning and a priori organization might be the efficient way to proceed.

But, what about the cost of the lost opportunity for developing valuable skills? A survey of students in Durango indicated one of every six students had access to a personal computer away from school. Should the lack of interest or economic resources for a family determine that their children have no opportunity to develop vital skills for the future? While it is true that not all students in the area are receiving instruction on microcomputers, many students have been given the opportunity because of the leadership and enthusiasm of a few key people in each school.

Nothing that is happening in southwestern Colorado is especially unique. Similar developments are happening throughout the United States and other parts of the world. The amazing realization is that developments are happening in schools that have been accused of not having had a new educational idea for several generations.

Does a school lead community consciousness or does it reflect the values of the community? In southwestern Colorado, this age-old question is being answered with a resounding "yes" to both parts in regard to microcomputers in our lives. Many schools and their communities are beginning to return to the viewpoint that the schools are a focal point for community aspirations and good will. If this trend continues, the microcomputer may be the instrument that brings people and their schools back together.

Anyone desiring further information may contact the author at Fort Lewis College.

MANAGING MICROMANIA AND BEYOND

Dr. Peter W. Wright
Edmonton Public Schools
10010 - 107A Avenue
Edmonton, Alberta, Canada, T5H 0Z8

ABSTRACT

This paper describes the approach used in a successful Apple based microcomputer inservice program for teachers - but will Apples always have "A Peel"?

INTRODUCTION

With more than 3000 teachers serving some 68,000 students in over 185 schools, Edmonton Public Schools is one of the largest school districts in North America. The District's involvement with the instructional use of microcomputer technology commenced in earnest at the beginning of the 1980-1981 school year with the implementation of a two year Microcomputer Pilot Project involving three schools and featuring eight Apple microcomputers. During the course of the Project, the number of microcomputers in the District increased from approximately ten to more than four hundred resulting in the provision of inservices (an already identified objective) becoming the focal point of Project activity.

This paper describes Edmonton Public Schools' successful, centrally administered microcomputer inservice program. In particular, the approach to delivery employed in the beginner level computer literacy inservice will be featured.

THE INSERVICE PROGRAM

The inservice program consists of three phases as outlined below where component services have been grouped within a phase according to the actual or projected needs of the District.

Phase 1 - Critical

a) computer awareness
b) BASIC literacy - beginner
c) applications software

Phase 2 - Highly Desirable

a) BASIC literacy - intermediate
b) subject related applications
c) instructional support applications
d) authoring systems
e) use of files

Phase 3 - Particular

a) BASIC literacy - advanced
b) authoring languages
c) high resolution graphics
d) beyond BASIC
e) word processing
f) advanced use of files
g) use of peripherals

The central activity has been the beginner level computer literacy inservice. This inservice, which presumes no prior knowledge, is a three, one half day experience of approximately 10.5 hours total duration. The major goal of the inservice is the achievement of both operational and programming oriented literacy within an educational context. A three (consecutively scheduled) half day format for this particular inservice maximises absorption and retention while minimizing the necessity to review.

A basic tenet associated with the beginner level inservice is that this first experience with computers should be a positive and successful one. For this reason, the inservice is typically a small group (5 - 7 participants), hands on, highly structured, one person per computer experience.

Operational literacy is achieved

through the use of curricularly specific applications software and accounts for approximately one third (the first half day) of the inservice time. The remaining two thirds of the inservice time is devoted to the achievement of programming oriented literacy through familiarization with approximately twenty five BASIC programming language commands. Most capabilities of the typical, school based, microcomputer system are highlighted in the inservice. Microcomputer learner outcomes have been documented to define the knowledge, skills and attitudes expected to be conveyed and/or developed as a result of this particular inservice.

THE APPROACH OF THE BEGINNER LEVEL INSERVICE

The First Half Day

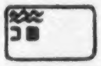
After outlining the basic objectives of the inservice, the first half day commences with a twenty minute overview of the computer system featuring very extensive use of analogies. This overview takes place with the lid of the computer removed and pays particular attention to the conception of memory and information flows in the computer environment - aspects which are considered to be critical to the understanding and successful operation of the computer. An example of the use of analogies is given here: Having distinguished between the two basic types of computer memory (external and internal) it becomes appropriate to distinguish between Random Access Memory (RAM) and Read Only Memory (ROM). These memories are likened to reference books each of which can contain great wisdom. RAM is likened to a student's lecture note book. The book's wisdom is placed there by the student and has value. This wisdom is not only frequently referred to but can also be enriched through revisions or additions. ROM however is likened to a bible or a dictionary. Again, a reference book but a book to which it is usually not appropriate or possible to add further wisdom.

Such an analogy provides a conception of ROM and RAM which participants can "hang their hats on" while endorsing the understanding of information flow paths (both ways for RAM - one way for ROM). In addition, this analogy draws attention to the contempt displayed by the computer towards user wisdom in contrast to its own inherent wisdom.

Having completed the overview, the participants are instructed how to turn on the computer and allowed to puzzle (as long as their patience lasts) as to why the computer appears to be doing nothing of great value despite all rational attempts at intervention. At this point, the true understanding of ROM is transcended and it can be forgotten (almost forever). Following a few words of comfort, a successful "boot" is achieved with a carefully pre-prepared program disc.

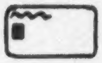
The unsuccessful "boot" and, in particular, the unsuccessful intervention attempts through the keyboard creates the opportunity to explain how one can determine how "ripe" the Apple is. The significance of the cursor and the language availability indicator are next explained. In this inservice, participants only come into contact with one language - Applesoft Basic. The two most probable screen displays are discussed in terms of what can be deduced about the state of "ripeness" of the Apple.

Most Probable Screen 1



Deductions:- 1. State of readiness
2. Applesoft in use
3. "Outside a program"

Most Probable Screen 2



Deductions:- 1. State of readiness
2. "Inside a program"

The meaning of outside and inside a program is described followed by appropriate courses of action consistent with the deductions. It is particularly important to point out that while the presence of the cursor indicates a state of readiness, the converse is also true. If the cursor is absent, the Apple is probably not ripe. At this point, the lack of success associated with the keyboard interventions during the unsuccessful boot will be understood. The computer simply was not in a state of readiness. Participants may recall that no cursor was present at that time.

The next step in the inservice is, perhaps, the most critical in terms of overall understanding. The way is now paved for understanding the onslaught of the INPUTS, PRINTS and CATALOGS - the FORS and RUNS and whatever NEXTS! A simple system has been devised which classifies computer commands into one of two types - Disc Operating System (DOS) or language and others. A simple rule is now needed to enable the distinction to be made. The disc drive has an "in use" indicator light. DOS commands access the drive. DOS commands thus cause the light to go on - others do not. The classification system thus used is:

DOS COMMANDS	LANGUAGE + COMMANDS
(light goes on)	(light doesn't)

Inservice participants classify the commands themselves. The mere existence of the classification system provides a focal point for understanding to take place.

The first half day proceeds with participants using programs of increasing complexity. The early programs are so simple that they allow an understanding of the computer and its information flows and the notion of being inside or outside a program.

By the end of the first half day, familiarity with a total of approximately eight operating system and language commands has been achieved, a foundation for future learning established and self sustaining user oriented literacy achieved.

The Second Two Half Days

The remainder of the inservice time is given over to a programming oriented approach to literacy. After first distinguishing between the immediate and deferred modes of execution, participants are taught how to write a simple two or three line program, edit it (by line deletion or re-definition - the poor man's editing) and to initialize a disc. Approximately seven programs will be written. The first takes a simple concept, such as squaring a number, and produces an application which accomplishes the task in a user friendly, aesthetic and discriminating manner. The majority of the language commands experienced in the inservice occur during the development of this first program which can take up to two and one half hours to produce. Throughout the inservice, emphasis is placed on exploiting acquired knowledge. For example, screens can be formatted entirely through the PRINT command if its potential is fully understood.

Participants are encouraged to "discover" how this can be achieved before being introduced to tabulation commands.

The desirability, and indeed necessity, of knowing a large number of commands is not disputed but the comfort level induced by showing how the potential of a few can be exploited in a variety of situations is valuable.

Subsequent programs developed in the inservice are smaller and designed to introduce concepts rather than produce live applications. Examples are programs which show that the computer is not restricted to dealing with numerical data and how one might control the degree of interaction with a particular application (e.g. loops and delays).

The first program developed revolves around the concept of squaring a number. Two stages in the development are outlined here to illustrate the approach to programming employed. Participants will have reached the following point.

```
10 INPUT A
20 PRINT A*A
30 END
```

This program functions but causes all inputs and outputs to appear on the screen. The challenge is to add one line containing one command to their program to achieve the desired result that a number and its square should be all that appear on the screen between successive uses of the program. It is very important that each individual works independently in addressing the problem. Participants are reminded that computers think and act quickly on instructions in a logical sequence. The essence here is that there are four places to incorporate the HOME command and in each case, the program works - it will square the number. All results, however, are different and only one is acceptable. Participants are directed to input a multi-digit number so that they can actually see that the program works (if they are quick). The message conveyed is that programming by trial and error is not very productive. The implication for the development of large programs barely needs to be stated.

At a later stage in the development of the first program, the following point is reached.

```
10 HOME
20 PRINT "PICK A NUMBER"
30 INPUT A
40 PRINT "THE SQUARE IS"
50 PRINT A*A
60 END
```

The challenge is to combine lines 40 and 50 and, using one PRINT command only, produce the following result:-

```
PICK A NUMBER
7
THE SQUARE IS 49
```

Participants are then left to test their understanding of the PRINT command in an essential period of what this author refers to as "floundering time". The results of this exercise are interesting and far reaching in their implications. The first time success rate is about 25%. The subsequent success rate on similar tasks is close to 100%.

The value of precise thinking is emphasised here through the computer's arrogant insistence on the correct use of the PRINT command.

Evaluation

An eight point questionnaire has indicated over 95% of participants to be highly satisfied with inservice activities.

BEYOND MICROMANIA

As the microcomputer bandwagon rolls along, rapidly gathering momentum, a number of interesting questions are raised. Are we for example, caught up in the euphoria of trying to make computer scientists out of our elementary school students and if we succeed, would it be an even greater triumph to accomplish such a result at the kindergarten level? An even more significant question is are we so preoccupied with the proliferation of microcomputers in education that we are not planning for the future use of the technology in general. It may well be that the true promise that the technology holds for education rests in its potential to support an individualized approach to learning.

A major initiative is presently being undertaken by Edmonton Public Schools to plan for the future use of computer technology in the District. A Distributed Systems Pilot Project is presently being pursued through which the integrated use of all levels of computer technology will be investigated within a network environment.

COMPUTERS IN SPECIAL EDUCATION:
A MODEL FOR MAKING THE INSTRUCTIONAL MATCH

Robert A. Zuckerman, Ed.D.
Assistant Professor of Special Education
Kent State University
401 White Hall
Kent, Ohio 44242

ABSTRACT: This paper describes a model for making the "match" between microcomputers and the instructional needs of special education students. A systems view of the learner is used to identify the hardware and software requirements best employed for a particular learner's instructional needs.

COMPUTER CAPABILITIES IN EDUCATION

Herriott (1982) identified seven functions that a computer can perform in education, which go beyond the limited view of the computer as the subject of instruction. These functions emphasize applications uniquely suited to instructional process in the classroom. Four additional functions not mentioned by Herriott have been included.

The first function identified is that the computer can be effectively used to "IMPART INFORMATION" to the student. The computer offers great flexibility for individualizing instruction as a "teaching machine." The computer has been shown to be very successful in conveying subject matter when the courseware has been well written and thoroughly validated. It is important to keep in mind that courseware development and validation is very time-consuming and costly and not much well-written, validated courseware is available. This is not to say that the abundance of "courseware" that is available is not and should not be used, but rather the outcomes of use may not be totally predictable in terms of the instructional intent/objectives established by the teacher.

The second function which the computer can perform is to "PROVIDE IMBEDDED REMEDIAL INSTRUCTION." Remedial instruction can be easily accomodated in courseware through the provision of additional practice problems and/or internal branching to review content not mastered or to present content not mastered in greater detail/concreteness. A particularly attractive feature of imbedded remedial instruction is that the student may not necessarily be aware that it is taking place. Hence, two students of differing abilities may be using the same instructional courseware without attention being drawn to the better or poorer student as may occur in the common example of grouping for instruction in reading or math or sending the student to a tutor/resource room/special class for help.

The third function is very similar to one just presented: The computer can "PROVIDE ENRICHMENT MATERIAL." The enrichment material may be imbedded within the courseware in the same manner as remedial instruction. The main difference would be that a student displaying mastery of content would begin to receive more difficult problems or internally branched to receive greater depth/breadth of the content being presented.

The previous two functions both emphasized instructional accomodations within the courseware primarily selected for imparting information. A function that should be noted is the use of the computer to REINFORCE AND SUPPORT TRADITIONAL MODES OF INSTRUCTION. This function relies upon initial instruction occurring in large or small group activities and the computer being used for specific enrichment and remediation purposes.

The fourth function identified is that the computer can "KEEP ACCURATE TRACK OF PROGRESS" of a student in an instructional program and in some instances across a courseware package. This is implimented either as a specific feature of an instructional program or an available feature of courseware developed under an authoring system. If all courseware used is developed under a specific system, then progress can be tracked across many instructional programs on varied subject topics.

The computer can only do what it is programmed to do. This means that a major factor in the flexibility of the computer is the program or courseware utilized. The fifth function that the computer can perform under program control is to "ALLOW THE STUDENT TO PROGRESS AT HIS/HER OWN RATE." This variable pacing that is student determined is important to individualization, but there is also the capability in the computer to be fixed paced.

To increase reading speed, response time etc. waiting until the student responds works against the development of "fluency-type" cognitive and motor skills. Hence another function available in the computer is to FIX/SET THE PACE OF PRESENTATION, time allowed for response, etc. in the courseware for the purpose of skill-speed development in students.

The sixth function that computers can perform in education are related to controlling a variety of output devices linked to the computer. The computer can "PROVIDE VIDEO AND AUDIO SUPPORT VIA PERIPHERAL DEVICES" to enhance instruction. This increases the likelyhood of being able to match the form of the output of the computer to the modality of reception of the student and hence raise the potential for instructional success.

Another function that the computer can perform in education is similar to the sixth. The computer SUPPORTS A VARIETY OF MODES OF INPUT from the student. A student can use a computer by voice as well as an almost infinite variety of motor movement responses. The variety of output and input modes available with computers has definite implications to use in special education settings.

The last function identified by Herriott is that the computer can "PROVIDE A MASSIVE INFORMATION RETRIEVAL BASE -- EITHER BY DIRECT DISPLAY OF THE MATERIAL ITSELF OR BY DIRECTING THE STUDENT TO THE APPROPRIATE MEDIUM."(p. 82) This function allows the student to employ the computer as a resource tool, the means by which the student selects material to view or obtains referral to a source.

One function overlooked by many is that the computer can PROVIDE A SOURCE OF MOTIVATION. Many devices like typewriters and calculators have been described as autoeletic, the mere presence of the device is the source of motivation to use it. On the one hand the computer provides the motivation for its use, but another interesting phenomena is that the computer can provide motivation for the development of reading, writing, and mathematics skills in students.

For educators to get the most out of using computers in the classroom they need to be aware of the functions it is capable of performing in the context of teaching/learning interactions. In addition to awareness, the educator needs to recognize the diversity of instructional applications and develop skills to selectively incorporate the computer when it is determined that the computer is an appropriate medium. By analyzing the available functions of computers in the context of the educational process, educators can better plan for use by matching the unique features of the computer to student and/or subject needs.

Computers have the ability to provide the educational functions described but only if appropriate hardware and software are present in the system. Each educational computer function requires hardware and/or software capabilities in the system to obtain the desired results in terms of function. In general, educational functions emphasizing input or output mode alternatives will rely on the fact that the appropriate hardware is present as well as the software to accomodate the various modes. All other educational functions may be addressed by careful selection of courseware.

The microcomputer holds unique promise for the special education class but only if not limited by the teacher. This limit can be lessened if the special education teacher recognizes how computer applications and the computer itself can address the needs and abilities of exceptional students. Consider the characteristics that are associated with various exceptionalities. For successful instruction of exceptional individuals to occur in the classroom, the characteristics present in the exceptional student must be accounted for in the subject matter being taught as well as the methods and materials employed by the teacher. Every college text about exceptional children provides some list of the characteristics that tend to be associated with various areas of exceptionality.

COMPUTERS AND THE EXCEPTIONAL STUDENT

How computers may be employed in education for the "mentally retarded" or "learning disabled" or any other exceptionality must be determined by the match of the computer's role, function, and mode of interaction with the student's needs and abilities. Just because one student identified as Learning Disabled benefits from a specific computerized instructional program does not justify its use for other students also so identified.

Not all the characteristics identified for a particular exceptionality can be addressed by the computer nor should they be. The computer should be employed when it can directly address an instructional skill or behavior in an effective and efficient manner. To attempt to prescribe computer use by exceptionality according to "handicap" will not address the individual differences inherent in students.

To use computers effectively and efficiently with exceptional children a match between the "learning skills" of the student and those aspects of the computer system that exist OR that are modifiable/adaptable must be made. The skills employed in learning can be viewed as INPUT to the student, PROCESSES within the student, and OUTPUT from the student. In the student, input overlaps with process and process overlaps with output. Inputs include the modalities by which the individual receives "data" (i.e., visual, auditory, kinesthetic, tactile) while outputs include the channels by which the individual expresses "data" (i.e., motoric-movement, oral-expression, non-verbal expression). Process in the student includes skills of perception and cognition of "data".

The critical factor in effective and efficient use of computers with exceptional students (and all other students for that matter) will be the match between the "skills" of the student and the educational function determiners of the computer. The match may be for the purpose of increasing a learning skill in a student by selecting a determiner that allows the student to develop and practice a new or weak learning skill OR for the purpose of maximizing content learned by selecting a determiner that is congruent with the student's personal learning skill strengths. Figure 1 depicts the student to computer interaction system and identifies the points in the system where matches must be made and match alternatives.

INSERT FIGURE 1.

Three types of matches must be made when planning for the instructional use of the computer with exceptional students. The primary match is that of the process skills of the learner to the cognitive demands of the courseware. This determination is made in terms of the instructional objectives identified for the student in the context of the curriculum and whether or not the content and process of the courseware adequately addresses those objectives.

The next match that should be made is between the computer output mode alternatives and the input skills of the student. For "visual learners" CRT or hardcopy from the computer are appropriate while auditory learners will benefit more from synthesized speech or audio tape controlled by the computer. The nature of visual computer output can be iconic (graphic) or digital (printed words) and hence may be further criterion for selection of the computer output/student input match.

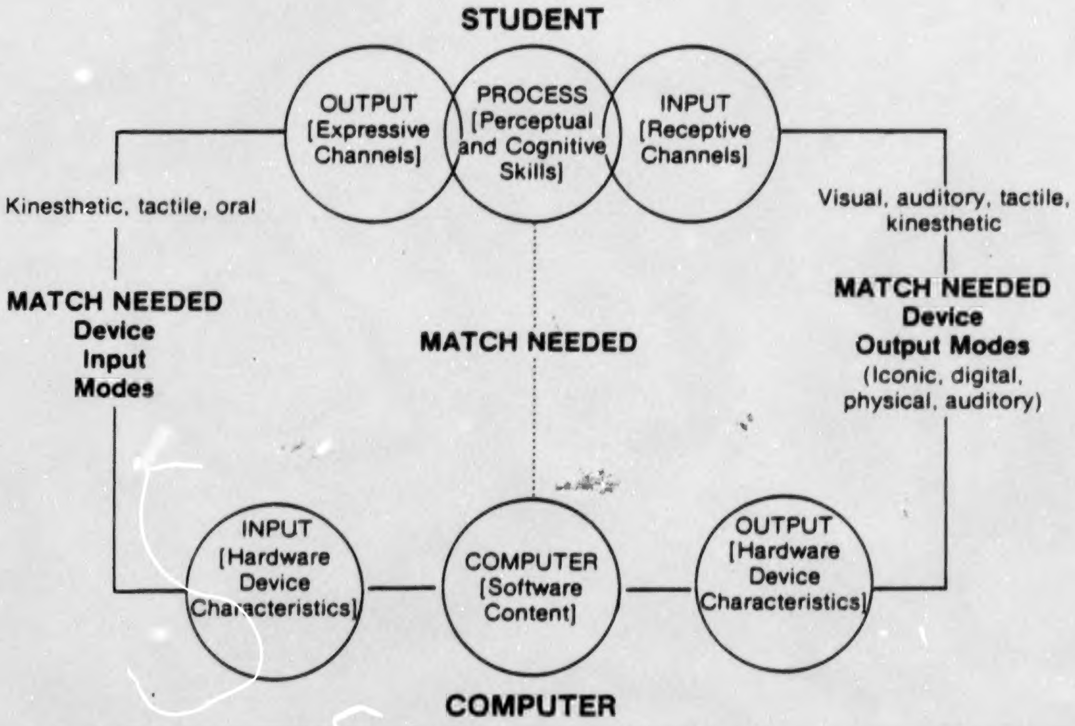
The final match to be made is between the computer input mode alternatives and the output skills of the student. For the physically impaired this match may need to be made prior to the other matches. Students without typing/spelling skills can make responses with joysticks, botton pads, light pens, touch switches, photo-switches, voice actuated response, etc. Those students with fine motor skills/manual dexterity can make their responses via qwerty keyboard or digit keypad. For some students fine motor skills can be developed and practiced by careful selection of a response mode that is to be emphasized as an instructional objective.

In the final analysis, the impact of microcomputers on education will depend on only two limitations during the next decade. The primary limitation of the microcomputer in education will be the limits imposed by the individual who plans for and/or uses it. The secondary limitation will rest in the hardware and software available.

REFERENCES

Herriott, John, "CAI: A philosophy of education and a system to match." In *Creative Computing* April, 1982, Vol. 8 No. 4, pp. 80-86.

Figure 1. STUDENT TO COMPUTER MATCH


STUDENT
OUTPUT [Expressive Channels]
PROCESS [Perceptual and Cognitive Skills]
INPUT [Receptive Channels]
Kinesthetic, tactile, oral
Visual, auditory, tactile, kinesthetic
MATCH NEEDED Device Input Modes
MATCH NEEDED
MATCH NEEDED Device Output Modes (Iconic, digital, physical, auditory)
INPUT [Hardware Device Characteristics]
COMPUTER [Software Content]
OUTPUT [Hardware Device Characteristics]
COMPUTER